



AFRL-RI-RS-TR-2014-102

SOFTWARE AND SYSTEMS PRODUCIBILITY COLLABORATION AND EXPERIMENTATION ENVIRONMENT (SPRUCE)

LOCKHEED MARTIN ADVANCED TECHNOLOGY LABORATORIES

APRIL 2014

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2014-102 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/ S /

WILLIAM MCKEEVER
Work Unit Manager

/ S /

MARK H. LINDERMAN
Technical Advisor, Computing &
Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE (DD-MM-YYYY) APRIL 2014		2. REPORT TYPE FINAL TECHNICAL REPORT		3. DATES COVERED (From - To) APR 2008 – SEP 2013	
4. TITLE AND SUBTITLE SOFTWARE AND SYSTEMS PRODUCIBILITY COLLABORATION AND EXPERIMENTATION ENVIRONMENT (SPRUCE)				5a. CONTRACT NUMBER FA8750-08-C-0064	
				5b. GRANT NUMBER N/A	
				5c. PROGRAM ELEMENT NUMBER 63781D	
6. AUTHOR(S) Srini Srinivasan, Scott Kagan, Douglas Schmidt, and Allison Park				5d. PROJECT NUMBER SPRU	
				5e. TASK NUMBER CE	
				5f. WORK UNIT NUMBER 08	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lockheed Martin Advanced Technology Laboratories 3 Executive Campus Cherry Hill, NJ 08002				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RITA 525 Brooks Road Rome NY 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI	
				11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2014-102	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. PA# 88ABW-2014-1525 Date Cleared: 09-Apr-2014.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This document describes the technical details of the Systems and Software Producibility Collaboration and Experimentation Environment (SPRUCE) project program execution during Phases 2, 3, and 4 spanning the period from April 2008 to September 2013. The SPRUCE was intended to facilitate the development of Software-Intensive Systems research products and methods, providing an environment for research of DoD systems and software problems, provide an ability for university and industry to leverage technology development, and establish a capability for successful technology transition and transfer.					
15. SUBJECT TERMS Software Collaboration, Software Experimentation Environment, Software Transition Technology, Software-intensive systems Producibility					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 69	19a. NAME OF RESPONSIBLE PERSON WILLIAM MCKEEVER
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) 315-330-2897

Table of Contents

TABLE OF FIGURES	III
1. SUMMARY	1
2. INTRODUCTION	3
3. METHODS, ASSUMPTIONS, AND PROCEDURES	4
3.1. PROGRAM SOFTWARE OUTPUTS	7
3.1.1. <i>SPRUCE Portal Software</i>	7
3.1.2. <i>Experimentation Infrastructure Integration Software</i>	9
3.1.3. <i>Affinity Technology Implementation Software</i>	9
3.2. PROGRAM DATA OUTPUTS	9
3.2.1. <i>Challenge Problems</i>	10
3.2.2. <i>Candidate Solutions</i>	10
3.2.3. <i>Experiments</i>	11
3.2.4. <i>Communities of Interest</i>	11
3.3. PROGRAM OUTCOMES AND SYNERGIES	11
3.3.1. <i>Technology Transition Experiment</i>	11
3.3.2. <i>NSF Cyber-Physical Systems Virtual Organization</i>	12
3.3.3. <i>Air-force Cyber Innovation Center</i>	12
3.4. SPRUCE MARKETING MATERIALS	13
3.4.1. <i>SPRUCE Flyer</i>	13
3.4.2. <i>SPRUCE Poster</i>	13
3.4.3. <i>SPRUCE Datasheet</i>	13
3.4.4. <i>SPRUCE Brochure</i>	13
3.4.5. <i>SPRUCE Information Pamphlet</i>	13
3.5. SPRUCE PUBLICATIONS AND PRESENTATIONS	13
3.5.1. <i>Collaboration Technologies Symposium</i>	13
3.5.2. <i>IEEE Aerospace Conference</i>	14
3.5.3. <i>Software and Systems Technology Conference</i>	14
3.5.4. <i>IEEE RTSS</i>	14
3.5.5. <i>CPS Week</i>	14
3.5.6. <i>INCOSE 2013</i>	14
3.5.7. <i>Crosstalk Magazine</i>	15
3.5.8. <i>Software Technology News</i>	15
3.5.9. <i>SPRUCE Webinars</i>	15
4. RESULTS AND DISCUSSIONS	18

4.1.	SPRUCE PORTAL METRICS.....	18
4.1.1.	<i>Data Metrics</i>	18
4.1.2.	<i>Traffic Metrics</i>	19
4.2.	PROGRAM REVIEWS	20
4.3.	LESSONS AND SUGGESTIONS	20
4.3.1.	<i>Lessons learned</i>	20
4.3.2.	<i>Suggestions for future</i>	21
5.	CONCLUSIONS	23
6.	REFERENCES	24
A.	APPENDIX.....	25
A.1	Challenge Problem List.....	25
A.2	Candidate Solution List.....	29
A.3	Experiment List.....	31
A.4	Community of Interest List.....	33
A.5	SPRUCE Flyer.....	35
A.6	SPRUCE Poster	38
A.7	SPRUCE Datasheet.....	40
A.8	SPRUCE Brochure	49
A.9	SPRUCE Information Pamphlet	54
	LIST OF ACRONYMS.....	63

Table of Figures

FIGURE 1: CURRENT TECHNOLOGY IDENTIFICATION, DEVELOPMENT, AND TRANSITION PROCESS	4
FIGURE 2: SPRUCE-ENABLED TECHNOLOGY IDENTIFICATION, DEVELOPMENT, AND TRANSITION PROCESS	5
FIGURE 3: KEY SPRUCE CONCEPTS	6
FIGURE 4: GROWTH OF CHALLENGE PROBLEMS.....	18
FIGURE 5: GROWTH OF NUMBER OF REGISTERED USERS.....	19
FIGURE 6: GROWTH OF UNIQUE VISITORS.....	19

1. Summary

This document describes the technical details of the Systems and Software Producibility Collaboration and Experimentation Environment (SPRUCE) project program execution during Phases 2, 3, and 4 spanning the period from April 2008 to September 2013.

The SPRUCE was intended to facilitate the development of Software-Intensive Systems research products and methods, providing an environment for research of DoD systems and software problems, provide an ability for university and industry to leverage technology development, and establish a capability for successful technology transition and transfer.

The Software and Systems Test Track (SSTT) Phase I activity was the precursor to the SPRUCE project and consisted of defining, developing, and documenting a Concept of Operations (CONOPS) and system architecture to meet the program objectives. In this phase, two competing teams developed their vision and approaches for the follow-on work. The LM ATL team was selected to be the execution team for the SPRUCE project.

SPRUCE Phase 2 was a 39-month program with an objective to build and deploy the infrastructure for the portal and experimentation facility, to validate the CONOPS with 'live', sample data, and to populate the initial set of data. This phase was quite successful, having achieved its goals well ahead of schedule, and having exceeded the targets for populated data and user-base. All the major elements of the portal and the experimentation facility were tested and deployed. Also, with representative data populated in SPRUCE, the team participated in a number of conferences and webinars designed to spread the awareness of SPRUCE among the community.

SPRUCE Phase 3 was a 15-month program with an objective to expand both the data populated in SPRUCE and to vastly grow community participation. The main idea was to use experts (or, community moderators) in specifically identified focus areas (Multi-core, Modeling and Cyber-Physical Systems) to both contribute challenges and also to solicit the community to contribute challenge problems. This phase achieved its goals in the populated data mainly through moderator-contributed content, but did not meet the targeted user registrations or anticipated community contributions. Following this experience, it was concluded that SPRUCE would be better hosted by a set of institutions perceived to be neutral, with significant amount of service to the broader software engineering community and more importantly, an existing strong user-base.

SPRUCE Phase 4 was a 9-month program designed to transition the portal operations to the CSIAAC (Cyber Security and Information Access Center) and the content development and moderation strategy to the SEI (Software Engineering Institute). This phase achieved its goal of smooth transition of the technical operations to the chosen institutions. However, the broader strategic direction and associated tactical approaches designed to build a vibrant community remain under constant consideration and experimentation.

Community Development was indeed singled out as a significant risk area from the start; it was anticipated that constant experimentation and evolution would be necessary for SPRUCE to succeed. The currently identified approach of minimizing user commitment to participation through smaller interactions (via blogs and curated content) presents a promising pathway, especially in conjunction with the lead execution team consisting of the CSIAC and SEI.

2. Introduction

The Software and Systems Test Track (SSTT) Phase I activity was the precursor to the Systems and Software Producibility Collaboration and Experimentation Environment (SPRUCE) project and can be considered the first phase of a multi-phase BAA program. The overall objective of the multi-phase program was to create and deploy an open collaborative research and development environment to demonstrate, evaluate, and document the ability of novel tools, methods, techniques and run-time technologies to yield affordable and more predictable production of software intensive systems.

The Systems and Software Test Track was intended to facilitate testing of Software-Intensive Systems Producibility research products and methods, providing an environment for research of DoD embedded systems and software problems, provide an ability for university and industry leverage of technology development, and establish a capability for successful technology transition and transfer. Challenge problems for the open experimental platforms were to be made accessible for all the research teams. This environment was to enable a full range of collaborative technology challenges, run-time platforms and applications, experiments, evaluations, and demonstrations.

SSTT Phase I consisted of defining, developing and documenting a Concept of Operations (CONOPS) and system architecture to meet the program objectives. The LM ATL team was selected as one of the two awardees in Phase I and the CONOPS document released by AFRL for SSTT Phase II (called S2PRUCE2) adopted a portion of the LM ATL Phase I team's CONOPS.

Following SSTT Phase I, the SPRUCE project's goals were stated as follows, and LM ATL's team was selected to execute the following-on phases:

The poor collaboration among people working across the technology maturity lifecycle has created a "valley of disappointment" where DoD programs fail to adopt advanced technologies, regardless of their inherent promise. A regime of ad hoc policies and procedures for transitioning software research into software practice in avionics and other domains has arisen for technology transitioning.

The Systems and Software Test Track will facilitate testing of Software-Intensive Systems Producibility research products and methods, provide an environment for research of DoD embedded systems and software problems, provide an ability for university and industry leverage of technology development, and establish a capability for successful technology transition and transfer.

3. Methods, Assumptions, and Procedures

A common refrain among practitioners of large-scale software intensive systems, which are very complex among many different dimensions, is that emerging and cutting edge software engineering tools, designed to address different pieces of their problem, work well on toy problems but do not scale to their system. An equally common refrain from developers of advanced software engineering tools and techniques is that their tools have been demonstrated to work well in application studies they constructed, but that the practitioners lack the resources and initiatives to apply their tools to actual systems. Meanwhile, the risks associated with software intensive systems continue to intensify with software becoming an essential part of modern Department of Defense (DoD) systems. For example, the General Accounting Office (GAO) has cited that the fraction of capabilities implemented in software in a typical avionics application has greatly increased—from 8% in the F-4 program (1960s) to 85% in the F-22 program (2000). Application of emerging and cutting edge software engineering tools and techniques is the only way to effectively manage the risks. But, how can one break the logjam between practitioners and researchers? SPRUCE is based on the premise that a widely available repository of well-defined, “at-scale” challenge problems, experiments and benchmarks are essential to bridge these two different worlds.

Figure 1 illustrates the divide in the current process within the DoD ecosystem for identifying, developing, and transitioning software producibility technology. Government personnel working DoD acquisition programs *coordinate* with government personnel working research programs to define software producibility problems and research agendas. The problems are then described and written into research programs’ Broad Agency Announcements (BAAs) and performers are asked to bid specific development and transition plans for software producibility solutions.



Figure 1: Current technology identification, development, and transition process

Software producibility researchers are then awarded contracts to develop their technology. Unfortunately, these researchers typically have little or no relationship with engineers in the program or domain from which their particular challenge problem is derived. While researchers strive to understand and incorporate deep, specific knowledge about a problem domain, they lack the necessary detailed program information. Researchers thus have little choice but to design and conduct experiments that are abstract and typically small-scale representations of the real challenge problem. These results may show the promise of the new technology, but leave a large “credibility gap” in the minds of program engineers about whether the results will transition into the real

problem domain. History indicates that it is hard to successfully bridge this gap, leading to the “valley of disappointment” shown in Figure 1. The ultimate success or failure of technology transition thus depends on the *ad hoc*, opportunistic transition process described above where serendipity of the right people being in the right positions is the primary enabler for success.

The primary goal of SPRUCE is to address the technology transition problem and bridge the “valley of disappointment” described above. SPRUCE emphasizes artifacts (e.g., sanitized DoD application software, computational resources such as specialized avionics processors and workflow management tools and services), typically provided in the context of challenge problems, and experimentation around them to create a common clearinghouse for program engineers and technology researchers to discover joint interests and form collaborations. We believe such collaborations on real world software producibility challenges and the associated experiments using realistic artifacts are the key to successful technology transition and, hence, have designed SPRUCE to provide a web-based portal and systematic process for initiating, sustaining and documenting such experimentation and collaborations.

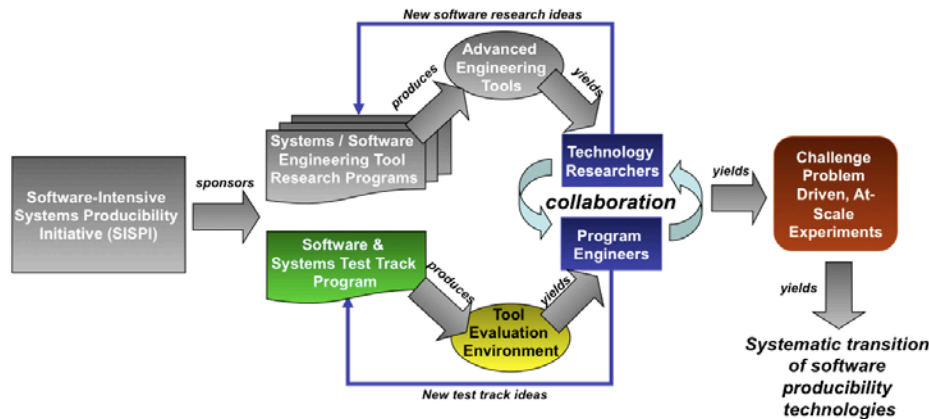


Figure 2: SPRUCE-enabled technology identification, development, and transition process

The SPRUCE collaboration environment, implemented as a web portal, seeks to empower its users to define and evolve narrow, well-defined technology problems of mutual interest, but at depth, and seeks to provide them with tools for collaboration and discovery. To achieve this goal, SPRUCE structures its collaboration environment around four basic concepts: communities of interest (CoI), challenge problems, candidate solutions, and experiments and experiment instances, as shown in Figure 3. These concepts are described below.

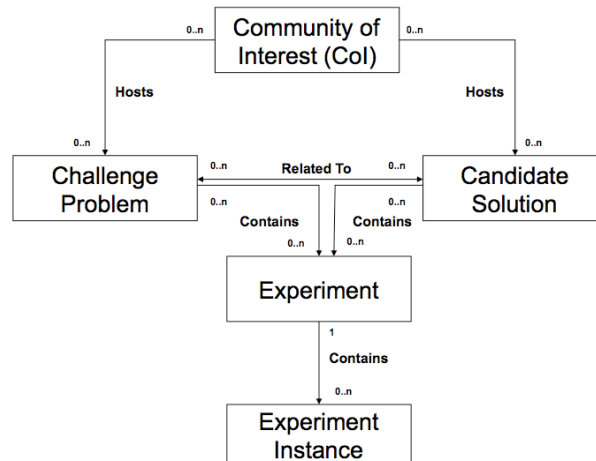


Figure 3: Key SPRUCE concepts

Communities of Interest (CoI): Communities of interest serve to organize SPRUCE content (i.e., challenge problems, candidate solutions and associated discussions) around broad but focused topic areas. They also serve as a virtual meeting place for SPRUCE users. SPRUCE users can belong to one or more communities of interest.

Challenge Problems: SPRUCE challenge problems represent sanitized versions of realistic problems that may occur on actual DoD acquisition programs. These problems may have occurred on other DoD programs in the past, may express a desire to solve future anticipated problems that would be tedious to solve using existing means, or may provide a context for radically new approaches to systems and software development. As these challenge problems represent a shared concern, they provide an opportunity to bring together the various stakeholders in the DoD software-intensive systems producibility (SISP) ecosystem.

SPRUCE encourages and enables DoD programs to submit realistic and sanitized artifacts that accompany challenge problems to attract researchers and provide real-world depth for challenge problems. The artifacts also present an opportunity for researchers to extend the provided artifacts along their areas of interest to highlight the applicability of their technology to the challenge problem provider in their own application context, or to the community at large.

Candidate Solutions: SPRUCE candidate solutions describe proposed solutions to SPRUCE challenge problems. Since SPRUCE challenge problems represent realistic problems faced by DoD programs, successful SPRUCE candidate solutions are more amenable to technology transfer. Researchers and tool vendors may, if desired, elect to upload their technology and tools into SPRUCE and to associate licensing conditions with the use of the tools. More likely, however, SPRUCE will be used to highlight specific properties of the tools and solutions and how they address specific challenge problems posed. Researchers and tool vendors can provide links to their solutions for interested SPRUCE users to access.

Experiments: SPRUCE experiments are associated with challenge problems and candidate solutions, and serve two primary purposes: (1) to showcase scenarios described in a challenge problem, so that SPRUCE community members have a repeatable baseline; or (2) to evaluate the effectiveness of a particular solution or set of solutions against a benchmark. In the former case, they are best initiated and mediated by the challenge problem provider, whereas a solution provider is best suited to define and conduct the latter kinds of experiments. *Experiment instances* represent an instantiation of a SPRUCE experiment that can be run on actual hardware, including the SPRUCE experimentation environment (discussed in the next section).

As shown in Figure 3, challenge problems, candidate solutions and experiments are interrelated and each can belong to one or more communities of interest. To facilitate a community's access to collaboration, SPRUCE automatically creates artifact repositories, community wikis, and discussion forums (termed 'collaboration items') for each of these entities and makes them readily accessible from the entity's main page. The use of social networking tools and instant communication facilities, such as rich text and media chat, as well as member presence information were being considered for future capabilities.

In addition to the web portal, SPRUCE provides an experimentation environment that is available to all SPRUCE users. This environment, comprised of real hardware resources, can be used to illustrate challenge problems and showcase candidate solutions in a repeatable manner on a representative environment. The SPRUCE experimentation environment is based on Emulab (www.emulab.net).

3.1. PROGRAM SOFTWARE OUTPUTS

Software was developed for various components of SPRUCE. Following are the software modules developed under the SPRUCE program

1. SPRUCE Portal Software
2. Experimentation Infrastructure Integration Software
3. Affinity Technology Implementation Software

Each of these is described in the following sections.

3.1.1. SPRUCE Portal Software

SPRUCE portal software implemented the majority of the SSTT CONOPS. This implementation is based on Microsoft Sharepoint software. Traditional spiral development was used. SPRUCE Phase 2 comprised of 3 spirals. SPRUCE Phase 3 consisted of one spiral. There were no software changes in SPRUCE Phase 4 because it was considered a transition phase. The following sections describe the various documents produced during the software development process.

SPRUCE Portal Software Requirements

One of the first documents produced initially was the requirements document. The purpose of this document is to record SPRUCE requirements that fulfill the use cases described in the Software and Systems Test Track Phase II CONOPS document [Concept

of Operations (CONOPS) for the Systems and Software Test Track Version 0.95], as well as additional use cases identified in the Lockheed Martin Advanced Technology Laboratories (LM ATL) SPRUCE proposal [Part I. Technical Proposal Systems and Software PRodUcibility Collaboration and Evaluation Environment (S²PRUCE²), July 19, 2007]. This document defined requirements for customizations to the collaboration portal platform. The experimentation portal and test bed were not envisioned to require customizations. This document mapped SPRUCE functional use case requirements to the technical features and development spirals of the collaboration portal. Spiral 1 implemented basic out-of-the-box (OOTB) capabilities, including those OOTB capabilities that were configured, to provide a rapid initial release cycle. Spirals 2 and 3 generally focused on features that required more substantial development effort to implement/realize.

SPRUCE Architecture Guide

The purpose of the Architecture Guide is to record all architectural information pertaining to the portal, including server locations, configurations, applications and the appropriate points of contact. This document contains all relevant server information including: IP addresses, applications running, make/model, specifications, and functions.

SPRUCE Portal Design Documents

SPRUCE Design Documents include the Entity Relationship Diagram, Data Taxonomy, Navigational Taxonomy and Permissions Taxonomy. In addition, the SPRUCE Data Model is also part of the design package.

SPRUCE Portal Re-design

In Phase 3, SPRUCE underwent a redesign, with significant changes to navigation and graphics. The graphics and navigation are part of the software deliverable and are included in the DVD media deliverable. Additionally completeness score was implemented during Phase 3. The formula for the calculation of completeness score is:

Following are on a scale of 1-5, with 5 being best:

- Problem Description: Weight: 5: Max Weight Value: 25
- Metrics: Weight: 5: Max Weighted Value: 25
- Artifacts: Weight 3: Max Weighted Value: 15
- Experiment: Weight 2: Max Weighted Value: 10
- Community Weight 5: Max Weighted Value: 25

Total max weighted value (Completeness Score): 100

Consider the following example case:

- Problem Description: Weight: 5: Example Case: 5, Weighted Value: 25

- Metrics: Weight: 5: Example Case: 3, Weighted Value: 15 (to improve, specify how the random and worst- case assignment are derived; also specify the improvement desired).
- Artifacts: Weight 3: Example Case: 3, Weighted Value: 9 (to improve, identify specific field with the artifact and how to use it in an experiment)
- Experiment: Weight 2: Example Case: 0, Weighted Value: 0 (to improve, identify the experiment)
- Community Weight 5: Example Case: 1, Weighted Value: 5 (to improve, identify collaborators)

Total: 54 (25+15+9+5)

Completeness Score: 54/100

SPRUCE Portal Transition

In Phase 4, the SPRUCE portal was transitioned to the Cyber Security & Information Systems Information Analysis Center (CSIAC) portal. During this process, we helped identify the elements that needed to be migrated, the elements that could be deferred and the elements that should not be migrated.

SPRUCE Portal Software Deliverables

Software developed under the SPRUCE program to support the portal operation is delivered in a DVD media; please note that a valid MS SharePoint instance is also required for the operation of this software.

3.1.2. Experimentation Infrastructure Integration Software

Software was developed under the SPRUCE program to interface Emulab (ISISLab instance) with the SPRUCE portal. Software associated with this integration piece is delivered in a DVD media. Please note that this feature is not supported in the transitioned CSIAC SPRUCE. It is included in this final report for completeness and future reference in the event such a feature is developed for the new SPRUCE portal.

3.1.3. Affinity Technology Implementation Software

Software was developed under the SPRUCE program for Affinity oriented searching of researchers and publications, when a challenge problems description is available. The software associated with this integration piece is delivered in a DVD media. Please note that this feature is not supported in the transitioned CSIAC SPRUCE. It is included in this final report for completeness and future reference in the event such a feature is developed for the new SPRUCE portal.

3.2. PROGRAM DATA OUTPUTS

Data results from the program include the basic SPRUCE elements described in section 3, and the collaboration data in the form of wiki, discussions, and artifacts. All of

this data is both migrated to the CSIAC, as well as delivered on a DVD Media. Following sub-sections present a summary of the basic elements and provide additional narrative insight, which might be useful future maintenance.

3.2.1. Challenge Problems

The current list of challenge problems in SPRUCE is in Appendix A.1. There are 81 challenge problems.

- The challenge problems that begin with the title “NRC Goal ...” represent text from National Research Council (NRC) Critical Code report. These challenge problems have no artifacts, experiments and metrics.
- Challenge problems that begin with the title “BAA-RIK-12-06 ...” represent technical areas from the Advance Software Engineering Technologies for the Software Producibility Initiative (ASETS) BAA. These challenge problems have no artifacts, experiments and metrics.
- The challenge problems that begin with “SOA-MANET ...” are placeholders for broad challenges in the Service-Oriented Architectures (SOA) and mobile ad hoc network (MANET). These challenge problems have no artifacts, experiments and metrics.
- “Cache False Sharing...” challenge problem represents a complete problem with description, artifacts, metrics, wiki, candidate solution, experiments and experiment instances.
- “Multi-dimensional Resource Optimization ...” challenge problem includes an artifact that lists the 10,000+ messages that are exchanged in an avionics application. There are other derivative challenge problems that incorporate the same title string.
- “Model Driven Architecture Design” challenge problem represents a large system of systems development for a Navy program. There are many artifacts attached to this challenge; also, there are many sub-challenge problems associated with this challenge as listed in “Related Challenge Problems” list.
- Challenge problems 22,23 (ID: 91,92) are from National Institute of Standards and Technology (NIST) Software Assurance Metrics And Tool Evaluation (SAMATE) program, and include rich set of sample source code artifacts.

3.2.2. Candidate Solutions

Current list of candidate solutions in SPRUCE is in Appendix A.2. There are 16 candidate solutions.

- “Deployment Automation Using Particle Swarms” candidate solution matches the “Multi-dimensional resource optimization ...” challenge. It also includes a rich set of experiments and experiment instances that can be run in the SPRUCE experimentation environment.

- “Genetic deployment ...” and “Hybrid particle swarm...” are related solutions to the above.
- The “Perseus” solution matches the “Cache-False Sharing ...” challenge.

3.2.3. Experiments

Current list of experiments in SPRUCE is in Appendix A.3. There are 15 experiments in SPRUCE.

- “Thread pairs ...” experiment uses the experimentation environment to illustrate the “Cache False-Sharing” challenge.
- 3 experiments with “Deployment Plan” in the title are related to each other and make use of the experimentation environment. They address the “Multi-dimensional Resource Optimization ...” challenges.
- The last 6 experiments (10 through 15) represent course materials and assignments from Vanderbilt University’s Dr. Gokhale’s courses.

3.2.4. Communities of Interest

Current list of communities of interest in SPRUCE is in Appendix A.4. There are 18 communities of interest (COI) in SPRUCE. COI’s serve as labels for challenge problems, candidate solutions and experiments. Registered users can associate themselves with different COIs. The CSIAC’s groups subsume the legacy SPRUCE COIs.

3.3. PROGRAM OUTCOMES AND SYNERGIES

3.3.1. Technology Transition Experiment

The SPRUCE project demonstrated its promise by bringing together, virtually, Lockheed Martin Aeronautics (LM Aero) and Vanderbilt University (VU) to collaborate around a multi-dimensional computing resource allocation challenge problem, representative of fighter programs such as F-22 and F-35.

The challenge problem posed is intractable in general, but has high potential payoff - reducing computing resources reduces weight, cooling and energy requirements, and leads to higher aircraft performance and increased operational range. SPRUCE enabled a sanitized but realistic dataset to be shared between the two parties, and also provided rich collaboration facilities including wikis, discussion boards and document exchange, which then helped guide virtual discussions about the underlying assumptions needed during technology application. VU demonstrated that a modified version of an optimization technology they previously developed could perform the desired optimization, facilitating consideration of the technology in the program.

More detailed information on this challenge and solution is included in an article published in Crosstalk Magazine [1].

3.3.2. NSF Cyber-Physical Systems Virtual Organization

The National Science Foundation (NSF) sponsored Cyber-Physical Systems Virtual-Organization (CPS-VO) is intended to serve the CPS program and to (i) facilitate and foster interaction and exchanges among CPS PIs and their teams; (ii) enable sharing of artifacts and knowledge generated by the projects with the broader engineering and scientific communities; and (iii) facilitate and foster collaboration and information exchange between CPS researchers and industry. This involved:

- creating and maintaining a web-based repository and collaborative platform to facilitate the open exchange of research results, tools, and educational materials among CPS researchers and the broader community;
- hosting tutorials and workshops to promote community interest, understanding, and the use of new methods;
- identifying effective mechanisms for technology transfer;
- creating a consortium of small businesses with interests in cyber-physical system innovations;
- collecting and disseminating cyber-physical system challenge problems from industry.

The SPRUCE program collaborated with NSF program managers in the early stages of CPS-VO definition to educate them on SPRUCE and offer any help, including offering an instance of SPRUCE for quick start. When the award was announced, we worked with the CPS-VO execution team to create a partition for them on SPRUCE and sharing data and user-base. However, since their program is targeted towards to CPS program with specific objectives, they developed their own infrastructure over time. The ideas and mechanisms of SPRUCE had an impact on the design of CPS-VO.

3.3.3. Air-force Cyber Innovation Center

The Cyber Innovation Center (CIC) anticipated that SPRUCE would be used on a frequent basis. They expected to host multiple Warfighter-Industry Collaboration Enterprise (WIC-E) events throughout the year (the current trend is one WIC-E per quarter). Their plan was to use SPRUCE in the lead-up to these events and the follow-up afterwards. They envisioned SPRUCE being used to send challenges and request for info to all participants (in preparation for the WIC-E, during the WIC-E, and following up afterwards). They were planning to use SPRUCE to help groups communicate, collaborate, and share information regarding specific challenges set forth in each WIC-E event.

The SPRUCE team created an instance of the portal for AF CIC use, and customized it for their user, after justifying to our Program Manager, our minimal additional effort to support these CIC events. To our knowledge, the anticipated WIC-E events did not materialize.

3.4. SPRUCE MARKETING MATERIALS

Two sets of marketing materials were developed during the course of the program. They are briefly described below and included in the Appendix. Many of the concepts (especially the problems addressed by SPRUCE) remain currently relevant and thus the content of these materials can be reused.

3.4.1. SPRUCE Flyer

An initial SPRUCE flyer, shown in Appendix A.5, was developed in SPRUCE Phase 2 to distribute at conferences and other events. 100+ such flyers were distributed.

3.4.2. SPRUCE Poster

A graphically oriented poster, developed during SPRUCE Phase 2 and shown in Appendix A.6, was displayed at conferences (e.g., RTSS, SSTC).

3.4.3. SPRUCE Datasheet

A datasheet describing each SPRUCE element, shown in Appendix A.7, was developed during Phase 2 for sending to specific individuals or institutions after initial contact at conferences. Dozens of such datasheet were distributed in electronic format and in the form of printed material.

3.4.4. SPRUCE Brochure

The SPRUCE brochure, shown in Appendix A.8, was developed in SPRUCE Phase 3, and incorporates matching graphics from the re-designed web site. This brochure was distributed at many conferences in printed form; it was also emailed in electronic form to potential SPRUCE participants and evangelists. 100+ brochures were distributed.

3.4.5. SPRUCE Information Pamphlet

The SPRUCE Information Pamphlet, shown in Appendix A.9, was developed in SPRUCE Phase 3, and incorporates matching graphics from the re-designed web site. The datasheet was typically distributed to interested parties after initial contact. Dozens of datasheets have been distributed.

3.5. SPRUCE PUBLICATIONS AND PRESENTATIONS

3.5.1. Collaboration Technologies Symposium

We participated at the Collaboration Technologies symposium (CTS) 2009 in March 2009. At this conference, we presented a paper and displayed a poster [3]. Our team won the Best Poster Award at this conference.

This was the first conference designed to showcase our collaboration approach and get feedback from the state-of-art practices at that time. We did not target getting any new user participants through this conference.

We were perceived as a novel approach to collaborative R&D at this conference. It should be noted that several collaborative portals of today (Innocentive, Kaggle, Challenge.gov etc) did not exist at that time.

3.5.2. IEEE Aerospace Conference

Our paper to the IEEE Aerospace Conference was well received and accepted [4]. This was the first conference where we participated with an explicit aim of attracting SPRUCE users. Our paper illustrated the SPRUCE concepts and a complete dataset designed to demonstrate what users could expect from SPRUCE.

Although we were able to attract some initial interest, a sustained user engagement proved challenging. We present more discussion on this experience in Chapter 4.3. We did not participate in this venue in subsequent years.

3.5.3. Software and Systems Technology Conference

We hosted a Birds-of-a-Feather (BoF) session at the Systems and Software Technology Conference (SSTC) 2010. About 20 people attended the BoF. We presented an overview of SPRUCE, and solicited participation. One SPRUCE user presented a challenge and solution set at this conference and uploaded them to SPRUCE. This was a challenge with a goal of reducing software complexity, illustrated with an example of a calculator implementation.

3.5.4. IEEE RTSS

We hosted a joint SPRUCE workshop with the NSF CPS VO at the IEEE Real-Time Systems Symposium (RTSS) conference in 2010 in San Diego, CA. Around a dozen participants attended our workshop.

Also at this conference, we announced and solicited a challenge competition whereby participants can upload a challenge problem to SPRUCE, with the winner getting to present the challenge and their related research to a DoD Program Manager. There were no submissions.

We did not participate in subsequent RTSS conferences.

3.5.5. CPS Week

We hosted a tutorial + BoF meeting in conjunction with NSF CPS VO at the CPS Week in 2011, held in Chicago, IL. About 20 people attended our session. We presented an overview of SPRUCE, and solicited feedback and discussions from the audience.

3.5.6. INCOSE 2013

We attended International Council on Systems Engineering (INCOSE) 2013 in Philadelphia, PA. We represented SPRUCE at the Lockheed Martin booth – by greeting visitors there and having SPRUCE marketing materials distributed at the conference.

Additionally we participated at the INCOSE vendor challenge and educated the participants on SPRUCE, and the idea of using an enhanced challenge problem to attract not just vendors – but also users and researchers. This is one of the suggestions for future work – to interact with such conference communities with a challenge problem repository - we discuss in section 4.3.2.

3.5.7. Crosstalk Magazine

In 2011, we published an article in the Crosstalk magazine [1], illustrating the challenge problem and solution pair populated in SPRUCE. This article also helped to spread the benefits of SPRUCE to potential users.

3.5.8. Software Technology News

In 2009, we published an article in Software Technology News through DACS [2].

3.5.9. SPRUCE Webinars

We conducted four Webinars hosted by the CSIAC (known as Data and Analysis Center for Software (DACS) for some time). Following are the abstracts of the webinar materials are available from their web site, as referenced below, and listed in reverse chronological order. The first webinar was an introduction to SPRUCE, while the next 3 were conducted by domain experts in specific areas, with a goal of attracting participation in SPRUCE. We typically saw a spike in user registrations at SPRUCE (10+) following each of these webinars.

1. <http://vimeo.com/38052031>

Model Based Systems Engineering: A solution to complexity or just a complex solution?

Model Based Systems Engineering (MBSE) has been around for decades and has enjoyed a considerable amount of success and acceptance in industry and academia. However, MBSE is not without its challenges, particularly with respect to its practical application to large scale system development. As we research solutions to these open MBSE challenges, it is important for us to evaluate the state of MBSE based on how much system complexity it reduces relative to how much complexity it adds to the system development process itself. This interactive virtual panel features leading MBSE experts from industry, academia and the government discussing this and other issues. Using real-world experience, backed by data artifacts and experiments from the research and application domains documented in the SPRUCE portal and elsewhere, the panelists will present their points of view and wrap up with an evaluation of the state of the MBSE practice and actionable ideas that you can start implementing today.

2. <http://vimeo.com/22025916>

SPRUCE Model Driven Architecture and Design

This webinar is a must-attend for those developing large scale system-of-systems or those engaged in research and development of tools and technologies for model-driven development of such systems.

3. <http://vimeo.com/12495485>

SPRUCE - A Case Study in Multi-Dimensional Resource Optimization using Program-scale Data, Candidate Solutions.

A Case Study in Multi-Dimensional Resource Optimization using Program-scale Data, Candidate Solutions, and Experimentation"
Presenters: Jonathan Preston and Russell Kegley, Lockheed Martin Aeronautics; Douglas Schmidt and Jules White of Vanderbilt University If you are a practitioner in Distributed Real-time and Embedded (DRE) systems, you already know that it is critical to manage system computing resources effectively. Have you ever wondered how this challenge unfolds for complex production DRE systems? In the first part of this two-part presentation we describe the challenges of system-wide computing resource optimization using sanitized, program-scale load data from a production avionics DRE system. This resource optimization problem stems from recent trends migrating away from legacy federated architectures to integrated computing architectures that combine multiple applications together on a single platform, instead of the traditional approach of allocating each application to a separate computing platform. In the second part of the presentation, we will describe a candidate solution to this problem using Particle Swarm Optimization (PSO), which tackles some aspects of this challenge, and describe its strengths and limitations. To demonstrate the capabilities of the PSO solution we will also highlight empirical results obtained from experiments conducted on a production avionics dataset. All the challenges, data artifacts, collaborations, candidate solutions, and experiments covered in this presentation are accessible from SPRUCE (Systems and Software Producibility Collaboration and Experimentation Environment). SPRUCE is an open web portal designed to bring together DoD software developers, users, and software engineering researchers by collaborating on specifying and solving software producibility challenge problems. SPRUCE emphasizes collaboration around well-defined challenge problems with project-specific artifacts representative of DoD projects, and experimentation for reproducing the stated problems, establishing benchmarks and evaluating solutions. SPRUCE also hosts an online-accessible experimentation facility that stores and replicates experiments easily. SPRUCE is funded by the Office of the Secretary of Defense (OSD) and supported and managed by the Air Force Research Laboratory (AFRL).

4. <http://vimeo.com/6625664>

SPRUCE - Bridging the Gap Between Research and Practice

"SPRUCE - Bridging the Gap Between Research and Practice"
Web Portal for the Collaborative Engineering of Software Intensive
Systems Producibility Challenge Problems and Solutions: (A research
initiative of the Air Force Research Laboratory)

Presenters: Patrick Lardieri and Rick Buskens, Lockheed Martin
Advanced Technology Laboratories If you are a program
engineer/manager in a software intensive system development, or if you
are a researcher working in software and systems engineering, this is an
event that you cannot afford to miss. Learn how engineers and researchers
now have access to the tools needed to reach out to each other, define and
refine challenge problems, develop solutions to these challenge problems,
and conduct experiments, all with the goal of helping DoD R&D through
open collaboration, thanks to the SPRUCE project.

Come learn about SPRUCE and how you can get involved.

SPRUCE (Systems and Software Producibility Collaboration and
Experimentation Environment) is an open web portal to bring together
DoD software developers, users, and software engineering researchers by
collaborating on specifying and solving software producibility challenge
problems.

SPRUCE emphasizes collaboration around well-defined challenge
problems with project-specific artifacts representative of DoD projects,
and experimentation for reproducing the stated problems, establishing
benchmarks and evaluating solutions. In this interactive demonstration and
presentation, we will illustrate SPRUCE's key features, including self-
organizing communities of interest (CoIs), dynamically evolving
challenge problems with accompanying artifacts, and built-in
experimentation facilities to reproduce the problems and evaluate solution
benchmarks. Finally, we demonstrate early experiences and results with
representative CoIs and challenge problems.

4. Results and Discussions

4.1. SPRUCE PORTAL METRICS

During the Program Execution, SPRUCE portal statistics were collected, analyzed and reported along two dimensions: Data and Traffic. At various times, targets were established for specific data and traffic metrics. Data metric targets, which are typically under our control, were achieved for the most part; traffic metric targets, which needed broader and stronger community participation to materialize, fell short of expectations. Following subsections discuss these two metrics and statistical profiles of each.

4.1.1. Data Metrics

Data metrics were collected, tracked and reported (in Quarterly Technical Reports) for SPRUCE elements (number of challenge problems, candidate solutions, experiments and communities of interest) for each of the quarters. Targets set for the number of challenge problems in Phase 2 and Phase 3 were achieved.

Figure 4 shows the graphical profile of the growth of challenge problems populated in SPRUCE. It should be noted that we met or exceeded the target every year; the 5th year target was set at 75-100 challenge problems.

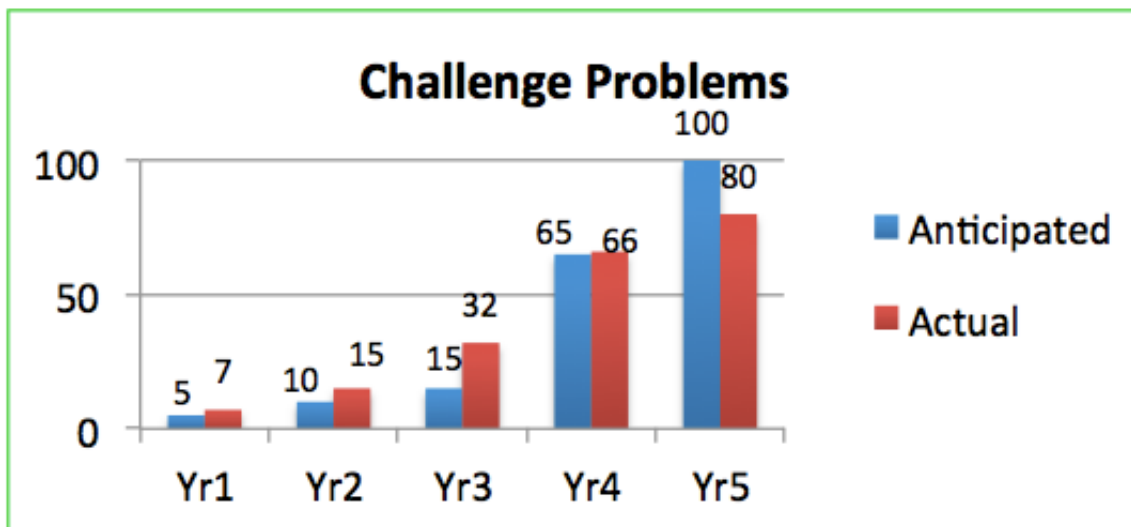


Figure 4: Growth of Challenge Problems

The final summary for the data metrics is shown in the following Table:

Measures of populated content in SPRUCE	Communities: 18 Challenge problems: 81 Candidate solutions: 16 Experiments: 15
---	---

4.1.2. Traffic Metrics

Traffic metrics were collected, tracked and reported (in Quarterly Technical Reports) for the number of registered SPRUCE users, number of visitors and the number of page views. Targets were set for the number of registered SPRUCE users in Phase 3. The growth in number of registered users is shown in Figure 5.

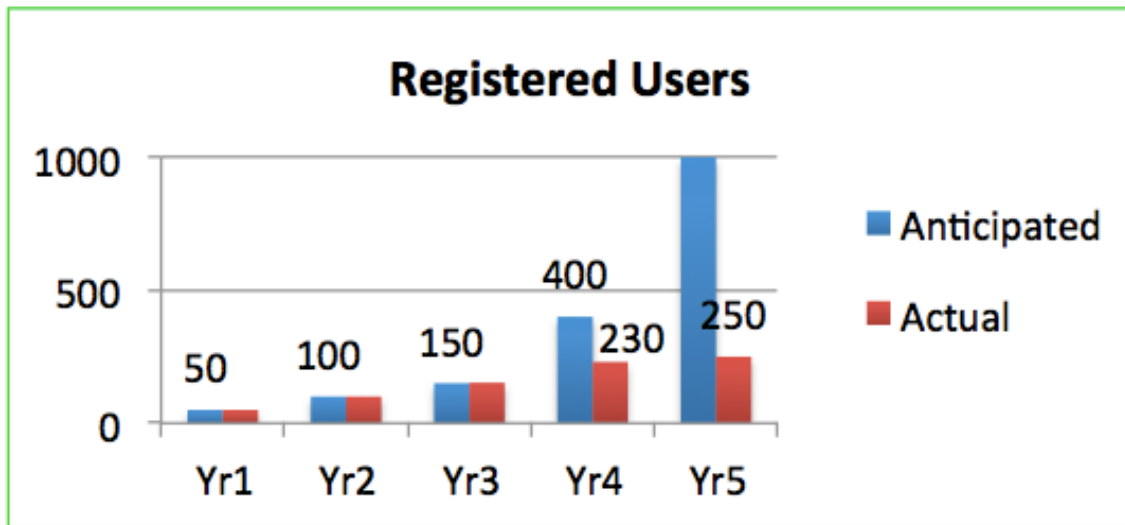


Figure 5: Growth of Number of Registered Users

The final tally for the number of registered SPRUCE users is 264. While this target did not materialize, we continued to see an increase in the number of unique visitors, as tracked by Google Analytics, and shown in Figure 6.

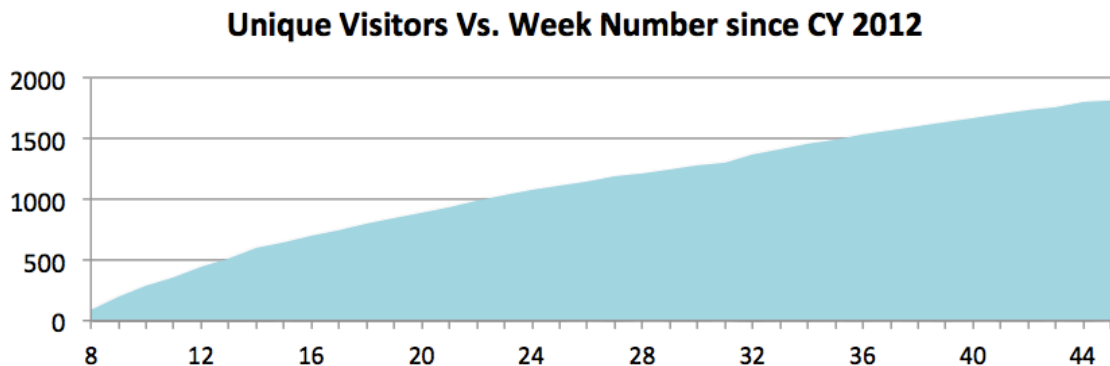


Figure 6: Growth of Unique Visitors

4.2. PROGRAM REVIEWS

We participated in several program review meetings during the program. We very much appreciated the discussions, feedback and assistance from the Program Managers and Program Sponsors during these meetings. These exchanges helped us partner with the Government and search for high-level strategic goals, set program targets, discuss tactical steps, as well as measure, discuss and act upon the results from those steps.

The presentation materials used at these meeting were submitted through usual channels per program requirements. For the sake of the brevity of this final report, the materials presented at these meetings are NOT included. Following are the list of review meetings during the SPRUCE program:

1. Phase 2 kick-off meeting, April 2008, Washington DC
2. Program review, November 2008, Cherry Hill, NJ
3. Program review, May 2009, Washington DC
4. Program review, November 2009, Nashville, TN
5. Program review, February 2010, Washington DC
6. Program review, September 2010, Forth Worth, TX
7. Program review, November 2010, Washington DC
8. SPRUCE demo and Program review, February 2011, Washington DC
9. Program review, May 2011, Rome, NY
10. Phase 3 kick-off meeting, September 2011, Via teleconference
11. Program review, November 2011, Washington DC
12. Program review, March 2012, Washington DC
13. Program review, June 2012, Washington DC
14. Program review, December 2012, Washington DC
15. Program review, June 2013, Washington DC
16. Final review, September 2013, Rome NY

4.3. LESSONS AND SUGGESTIONS

4.3.1. Lessons learned

We had identified community participation as the primary (“red”) risk for the SPRUCE project in our Phase 2 proposal. Building the portal, populating initial content and educating potential users were estimated as “low-risk” activities. This turned out to be true during the program execution.

With the benefit of hindsight, we are able to conclude that Challenge problems, Candidate solutions and Experiments require a lot of commitment in time from the user – hence an external, explicit incentive is needed for participation (e.g., anticipated high-probability research funding, explicit requirement from a program to participate in SPRUCE). Implicit incentives such as recognition from a community such as contributions to kernel.org or other open source movement will only work with very wide communities – reaching into hundreds of thousands or millions. This is a single most important barrier to participation – users being able to justify for themselves the effort required for contribution weighing the benefits of such contribution; we term this as the participation Return on Investment (ROI) barrier. Other issues such as International Traffic in Arms Regulations (ITAR), Intellectual Property IP etc., are important (these issues can be overcome using technology) but not as critical as the participation ROI. Some ways to overcome the participation ROI barriers are discussed in section 4.3.2.

Webinars are a low-overhead way of attracting participation and reaching a wider global audience. An archived version of the webinar can also be used as a persistent content as a cross reference from blogs and discussions to build a stronger Internet presence. Webinars exploring technical topics that point to SPRUCE, and those that use multiple technical experts that reference content in SPRUCE, but focus on the technical areas, attract more participation, engagement and awareness of SPRUCE than webinars educating users on SPRUCE and its benefits.

Paper presentations on SPRUCE and conference talks/posters designed just to market SPRUCE are not beneficial from an SPRUCE ROI perspective. Instead, it will be useful to encourage our moderators or technical experts to attend a domain-oriented conference (e.g., IEEE MODELS for Model-Driven Development) to present a specific challenge and/or related solution relevant to the Community of Interest, with references to SPRUCE and encouraging the audience to access the portal.

Education market is a good opportunity, but simply putting references to SPRUCE in online course materials does not encourage participation; need to specifically invoke participation from users – downloading assignments or data, requiring them to add to solutions, challenges or experiments.

Kaggle, Innocentive, and Challenge.gov are newer sites that have gained traction and content in the last several years while SPRUCE was operational. They use the concept of prize money for winners and hence provide incentive for solution providers to participate – thereby also providing incentive for challenge providers to participate. With our contract, we could not test this avenue –awarding prize money for winners; future contracts need to leave this option open for consideration by execution teams.

4.3.2. Suggestions for future

One of our suggestions from the Phase 3 led to SPRUCE being transitioned to neutral institutions in Phase 4. Both the Software Engineering Institute (SEI) and CSIAC are neutral participants in the DoD ecosystem, not aligned with any specific Systems Integrator. Moreover, they have a history of supporting the larger community over decades and support user-bases of tens of thousands of users each, which is two orders of magnitudes better than SPRUCE. With SPRUCE hosted at these institutions, DoD will be

able to leverage its past investments in these institutions and SPRUCE - and realize the vision of increased interactions leading to better transition opportunities.

Social networks and virtual collaboration facilities help reduce the friction that traditionally has existed between researchers and practitioners in Software and Systems Engineering. Thus these mechanisms should be part of the Government's strategy to increase the efficiencies of the technology transition process. Challenge Problem-centric active collaboration will make the transition process efficient, since its use-cases or CONOPS are closest to what the actual transition process is. However, as noted before, the challenge problem-centric collaboration suffers from the participation ROI barrier from the user's perspective. There are two ways to bring down the barrier: reduce the effort needed to participate, and make the benefits immediate and apparent.

To reduce the effort needed for participation in SPRUCE, lightweight interactions can be utilized. Examples are the blogs, webinars and articles written by experts or with curated content that can invite community participation, that can then lead to a well-defined, community developed challenge problem. With well-established and popular micro-blogging platforms of Twitter, Facebook and LinkedIn today, these user interactions represent low-effort means for users to start getting involved. A well-intentioned technical moderator can nudge these interactions towards actionable, fundable and collaborative effort. At the same time, social networks present immediate and apparent benefits to the participants, an opportunity for showcasing their thought leadership in a peer community. The combined SEI-CSIAC team is already exploring and experimenting with such an effort.

To improve the ROI for the Government, newer developments in social networking and crowdsourcing can be leveraged. Crowdsourcing platforms such as Kaggle and Innocentive are gaining popularity among the non-traditional participants in the DoD eco-system. These present an easy way to reach a broader audience and make them aware of the technical challenges that DoD is engaged in. For example, Innocentive sports a NASA Pavilion showcasing space-related challenges that the community can download and try to solve. Our prior interactions with Kaggle indicated that one can expect 100,000 visitors to a challenge problem, 1000 downloads and 50+ teams offering solutions. These numbers are 2 orders of magnitude better than what could obtain from an organic development of the SPRUCE portal, but incur a cost of approximately \$20K of prize money plus some overhead of the sponsoring platforms. Future contracts should be structured to enable execution teams to leverage these platforms. References and link-backs from a sponsored problem can be used to maintain and build 'brand' awareness.

5. Conclusions

The SPRUCE was intended to facilitate the development of Software-Intensive Systems research products and methods, providing an environment for research of DoD systems and software problems, provide an ability for university and industry to leverage technology development, and establish a capability for successful technology transition and transfer. This report covered Phase 2 thru phase 4.

SPRUCE Phase 2 was a 39-month program with an objective to build and deploy the infrastructure for the portal and experimentation facility, to validate the CONOPS with ‘live’, sample data, and to populate the initial set of data. This phase was quite successful, having achieved its goals well ahead of schedule, and having exceeded the targets for populated data and user-base. All the major elements of the portal and the experimentation facility were tested and deployed. Also, with representative data populated in SPRUCE, the team participated in a number of conferences and webinars designed to spread the awareness of SPRUCE among the community.

SPRUCE Phase 3 was a 15-month program with an objective to expand both the data populated in SPRUCE and to vastly grow community participation. The main idea was to use experts (or, community moderators) in specifically identified focus areas (Multi-core, Modeling and Cyber-Physical Systems) to both contribute challenges and also to solicit the community to contribute challenge problems. This phase achieved its goals in the populated data mainly through moderator-contributed content, but did not meet the targeted user registrations or anticipated community contributions. Following this experience, it was concluded that SPRUCE would be better hosted by a set of institutions perceived to be neutral, with significant amount of service to the broader software engineering community and more importantly, an existing strong user-base.

SPRUCE Phase 4 was a 9-month program designed to transition the portal operations to the CSIAC (Cyber Security and Information Access Center) and the content development and moderation strategy to the SEI (Software Engineering Institute). This phase achieved its goal of smooth transition of the technical operations to the chosen institutions. However, the broader strategic direction and associated tactical approaches designed to build a vibrant community remain under constant consideration and experimentation.

Community Development was indeed singled out as a significant risk area from the start; it was anticipated that constant experimentation and evolution would be necessary for SPRUCE to succeed. The currently identified approach of minimizing user commitment to participation through smaller interactions (via blogs and curated content) presents a promising pathway, especially in conjunction with the lead execution team consisting of the CSIAC and SEI.

6. References

1. Schmidt, D. C., Kegley, R., Preston, J., Dougherty, B., & White, J. (2011). Deployment Optimization for Embedded Flight Avionics Systems. CrossTalk: The Journal of Defense Software Engineering, Nov/Dec 2011, Vol. 24 No. 6
2. Patrick Lardieri, Rick Buskens, Srini Srinivasan, William McKeever, and Steven Drager, SPRUCE: Systems and Software Producibility Collaboration and Experimentation Environment, Software Tech News, April 2009 Vol. 12 No.1
3. Patrick Lardieri, Rick Buskens, Srini Srinivasan, William McKeever, and Steven Drager, SPRUCE: A Virtual Collaboration Portal for the Systems and Software Producibility Community, The 2009 International Symposium on Collaboration Technologies and Systems (CTS 2009) May 18-22 2009, Baltimore MD
4. Patrick Lardieri, Rick Buskens, Srini Srinivasan, William McKeever, and Steven Drager, SPRUCE: A Web Portal for the Collaborative Engineering of Software Intensive Systems Producibility Challenge Problems and Solutions, IEEE Aerospace Conference, March 2010, Big Sky Montana

A. APPENDIX

A.1 Challenge Problem List

[3 page]

Num ID	Challenge Problem Title
1	44 Cache False-Sharing in Multi-Core Architectures
2	50 Assessing the Evolution of Software Features
3	51 Distributed Integration strategies for Functional Components on MultiCore Architectures
4	56 Multi-dimensional Resource Optimization for Publisher_Subscriber-based Avionics Systems
5	59 Finding the functionally representative areas of a software system
6	61 Test S3 Challenge Problem
7	62 Advanced Multi-dimensional Resource Optimization for Publisher_Subscriber-based Avionics Systems
8	64 SOA-MANET Distributed Sensors
9	65 SOA-MANET Third-Party Message Translation
10	66 SOA-MANET Geographic Subscriptions
11	67 SOA-MANET Provisioned Workflow for combined SA picture service
12	68 SOA-MANET UCAV Fire Support
13	71 SOA-MANET Service Survivability and adaptation
14	72 SOA-MANET Topology Resilience
15	74 SOA-MANET Mission-Sensitive Networked Services
16	84 Adv. Multi-dimensional Resource Optimization for Publisher-Subscriber-based Avionics Systems Incl Pwr Consumption
17	85 Technology Readiness Assessment Management Software
18	86 Ility-related Risk Identification Management Software
19	88 FibreChannel Protocol Optimization in Real-time Embedded Systems
20	89 Reduce Software Overhead in Event Driven Systems
21	90 Model Driven Architecture and Design
22	91 Determination of Ground Truth in Large Scale Benchmarks for Evaluating Static Analysis Tools
23	92 Automated Test Code Generation for Static Code Analysis
24	93 Minimize re-testing in the face of multiple expected changes
25	94 Minimize testing across a product line of software
26	95 End-to-end Worst Case Response Time under Period Scheduling
27	96 Verification by Quantum Simulation
28	97 Measure and report on the ability of the system to meet performance objectives through system development lifecycle

29 103 Identify instances of incomplete or imprecise specifications in design models
 30 104 Enforcement of cultural knowledge and architectural vision
 31 105 Supporting Concurrent Development
 32 106 Tighter coupling between system architecture, vision and requirements
 33 107 Inconsistent Requirements Interpretation
 34 109 Automatic Detection of Deadlocks and Race conditions in Real-Time Systems Software
 35 110 Automatic Detection of Priority Inversions in Real-Time Systems Software
 36 111 Verifying Message Sequences in Real-Time Systems Software
 37 112 Real-Time Scheduling for Heterogeneous Workloads
 38 113 Cache management in multi-core systems
 39 114 Real-Time Scheduling in Heterogeneous Multi-Core Processors
 40 115 High-Utilization Multicore Global Scheduling Analysis
 41 116 Identify Insider Threat Controls
 42 117 Assessing the Generality and Reusability of a Capacity Planning Design Process for Cyber Physical Systems
 43 118 Enabling multi model based design and analysis
 44 120 Information Superiority at the Edge
 45 121 Capturing scalar data interdependencies between models in a multi-model
 46 123 Resource Optimization for Mobile Platforms at the Edge
 47 124 User-Controlled System Adaptation at the Edge
 48 125 Early Warning of Insider Threats
 49 126 Capturing conceptual interdependencies between models in a multi-model
 50 127 Identification and propagation of changes in scalar data interdependencies between models in a multi-model
 51 128 Identification and propagation of changes in conceptual interdependencies between models in a multi-model
 52 129 Tracking and managing model specific data tolerances for scalar data interdependencies
 53 130 Identifying and semi-automatically resolving cycles in model interdependencies
 54 131 Maintaining semantic integrity of scalar data interdependencies of models in a multi-model
 55 132 Maintaining semantic integrity of conceptual interdependencies of models in a multi-model
 56 137 High Confidence and Trustworthy Systems for Advanced Manufacturing
 57 138 High Confidence System Solutions for Maintaining Patient Normothermia During Perioperative Periods
 58 139 Determine schedulability and priority assignment for real-time tasks

59 141 BAA-RIK-12-06 Distributed Processing and Multi-core Processing
 60 142 BAA-RIK-12-06 Completeness and Development and Testing and Sustainment
 61 143 BAA-RIK-12-06 Instrumentation and Monitoring
 62 144 BAA-RIK-12-06 Legacy Software
 63 145 Automatic syntactic model transformation definition
 64 146 Model Based Systems Engineering - A solution to complexity or a complex solution
 65 147 BAA-RIK-12-06 Advanced Software Engineering Technologies for Software Producibility
 66 149 NRC Goal 1.1. Facilitate Mission-Oriented Modular Architectures
 67 150 NRC Goal 1.2 - Facilitate Architecture-Aware Systems Management
 68 151 NRC Goal 1.3 - Facilitate Architecture-Driven Development
 69 152 NRC Goal 1.4 - Facilitate Architecture Recovery
 70 153 NRC Goal 2.1 - Effective Evaluation for Critical Quality Attributes
 71 154 NRC Goal 2.2 - Assurance for Components in Large Heterogeneous Systems
 72 155 NRC Goal 2.3 - Enhance the Portfolio of Preventive Methods to Achieve Assurance
 73 156 Managing a Messaging Behavior Model in a Component-Based Architecture
 74 157 Measuring Architectural Technical Debt
 75 158 Estimation of Best- and Worst-Case Performance of Task Placement with Simultaneous Multithreading
 76 159 NRC Goal 3.1 - Enhance Process Support for Both Agile and Assured Software Development
 77 160 NRC Goal 3.2 - Address Supply-Chain Challenges and Opportunities
 78 161 NRC Goal 3.3 - Facilitate Application of Economic Principles to Decision Making
 79 162 NRC Goal 3.4 - Develop and Apply Policy Guidance and Infrastructure for Conducting Evidence-Based DoD Milestn Rvw
 80 163 NRC Goal 3.5 - Enhance Process Support
 81 164 Inverse Design Engine

A.2 Candidate Solution List

[1 page]

Num	ID	Candidate Solution Title
1	36	Using algebraic manifolds to assess feature evolution
2	39	Perseus
3	40	Deployment Automation Using Particle Swarms
4	41	Genetic Deployment Optimization
5	44	Test S3 'Solution'
6	45	Hybrid Particle Swarm / Bin-packing Deployment Solver
7	46	Open CPI
8	62	Technology Maturity Level Calculator
9	63	Risk Identification Integration and Ilities Calculator
10	64	Technology Maturity Calculator
11	65	Breaking the Time Barrier
12	66	Combinatorial design techniques
13	67	Software visualization tool
14	68	Visualization approach to Problem 3
15	70	Priority Assignment Schemes
16	78	Technical Engine for Democratizing Multiple-Domain Modeling and Simualtion

A.3 Experiment List

[1 page]

ID	Experiment Title
1	Thread Pairs Example of Cache False Sharing in Multi-Core Architectures
2	Analysis of GAIM and Firefox
3	Illustrate scenarios with priorities, when system is no longer manually scheduled
4	Sample Particle Swarm Generated Deployment Plan
5	Sample Deployment Plan from a Genetic Algorithm
6	Particle Swarm Deployment Optimization on ISISLab
7	Perseus
8	Reduced Complexity 5-Function Calculator
9	Incomplete Specification Illustration
10	Compare performance of different server architectures
11	Measuring overhead and jitter
12	Measuring the latency and jitter in demultiplexing and dispatching operations
13	Software Engineering Design Patterns Programming Samples
14	Using Real-Time CORBA
15	Using Real-Time Event Channels

A.4 Community of Interest List

[1 page]

ID	Community of Interest Title
1	Architecture-Based Testing
2	BAA-RIK-12-06 Advanced Software Engineering Technologies for the Software Producibility Initiative
3	Computer Science Education
4	Feature-oriented Software Analysis
5	Insider Threat Resistant System Design - Analysis - Operations
6	Multi-Core Architectures
7	Multi-model based Systems Engineering
8	Networked Systems of Systems
9	NRC Critical Code Report
10	OpenCPI or Open Component Portability Infrastructure
11	Real-Time and Embedded Systems
12	Reduced Complexity Software
13	Software Design Patterns
14	Software Systems Stockroom for Autonomous Flight Systems
15	SOS Architecture - Systems Engineering - Software Design
16	SPRUCE Portal
17	Static Code Analysis
18	Transition of Software Engineering Research to Practice

A.5 *SPRUCE Flyer*
[2 pages]



Systems & Software Producibility Collaboration and Experimentation Environment (SPRUCE)



SPRUCE BETA
Systems and Software Producibility
Collaboration and Experimentation Environment

This Site:

Sign In | Register | Sign Out | Feedback | Invite | Share

Home | Communities of Interest | **Challenge Problems** | Candidate Solutions | Experiments | Usage Metrics | Tools

Challenge Problem Details

Select a Different Challenge Problem...

Title: Cache False-Sharing in Multi-Core Architectures

Description: General Problem: Commercial multi-core architectures (e.g., Intel, AMD, PowerPC) are based on cache-coherent shared memory. Coherency between data held in per-core (L1) caches and per-processor (L2) caches is typically achieved via hardware-level protocols (e.g., MESI) that transparently invalidate and reload replicated cache lines as data is touched. A potential side-effect of keeping multiple copies of cached data coherent is the phenomenon known as "false-sharing". False-sharing occurs when multiple threads that are running on separate cores are accessing data that resides in the same logical cache line (typically 64 bytes in size) that exists as multiple copies across multiple caches. Each thread does not have to access the exact same memory location, but only locations that reside in the same logical cache line. For example, globally declared variables are often mapped in this manner. As each thread's cache associated with other cores are invalidated, the pattern effectively renders the cache ineffective since cache hits are continuously being reloaded and thus the advantages of data being in the cache in the first place is lost. The problem of false-sharing is exacerbated when thread locations change during the course of program execution (i.e., they are not pinned to specific cores). This problem can lead to sporadic performance degradation depending on the particular selections made by the operating system scheduler at run-time. The problem of false sharing can be avoided by either serializing access to data that resides in the same logical cache line or by spacing data layout in memory (i.e., padding) so that there is no possibility of multiple threads accessing data in the same cache line. Serialization can lead to dramatic reductions in

Actions

- Contact Sponsor
- Edit Descriptive Information
- Propose Challenge Problem
- Propose Candidate Solution
- Propose Experiment
- View Challenge Problem Scores
- Express Interest

Discuss

Wiki Entry

SPRUCE is sponsored by the OSD and managed by the AFRL Rome Labs. Please visit <http://www.sprucecommunity.org> to learn more and get your account.



Booz | Allen | Hamilton



Raytheon

SPRUCE

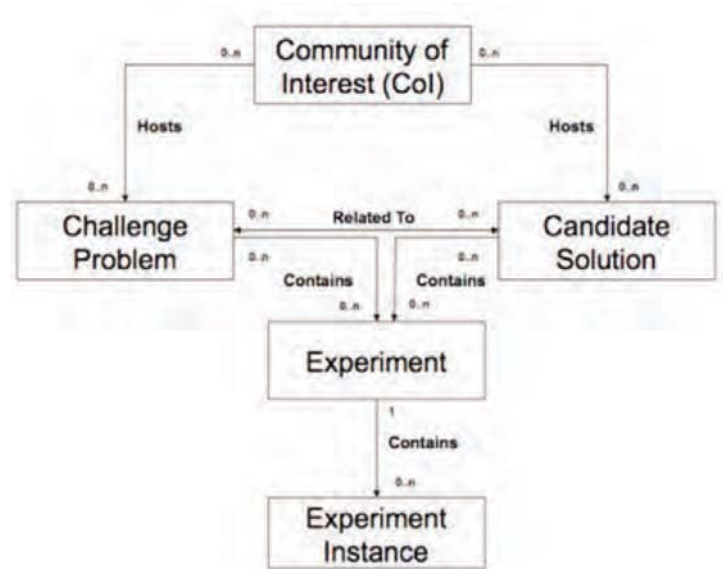
Revolutionizing the way that systems and software engineering technologies are identified, developed and evaluated

Features

- Customized collaboration portal
- Dedicated on-demand experimentation facility
- Repository of realistic problem artifacts
- Experiments to consistently reproduce problems and demonstrate solutions

Repository Elements

- **Community of interest:** virtual team of experts and practitioners to identify problems and work on solutions
- **Challenge problems:** sanitized versions of realistic problems and data that occur on DoD acquisition programs
- **Candidate solutions:** describe proposed solutions to challenge problems
- **Experiments and experiment instances:** show-case challenge problems and evaluate solutions



SPRUCE Can Help ...

Program Engineers

- Discover and reach out to a broad community that can relate to the same technical problems
- Learn about technologies relevant to the problem
- Engage a community of experts to solve the problem
- Make available sanitized, at-scale program data and demonstrate specific problems in the experimentation facility

Software Researchers

- Gain key insights into DoD problems
- Access a repository of real-world problems and representative data
- Quickly and easily engage an active practitioner community
- Effectively demonstrate the technologies and tools in the experimentation facility using program data

Post Challenge Problems



For best results, include at-scale, sanitized data, and reproducible experiments

SPRUCE Experiment Collaborate



Demonstrate Solutions



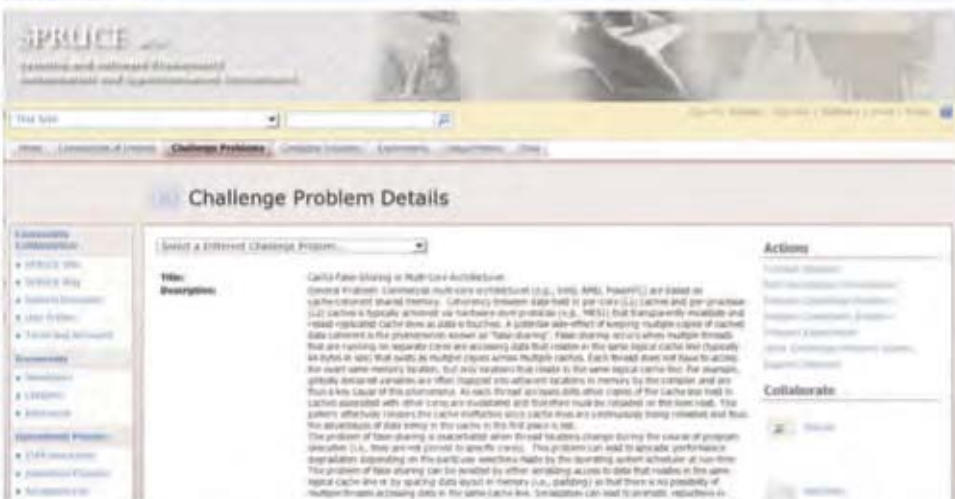
For best results, fully use the provided data and reproducible demonstrations

Visit www.sprucecommunity.org to learn more and get an account

A.6 *SPRUCE Poster*
[1 page]



Systems & Software Producibility Collaboration and Experimentation Environment (SPRUCE)



SPRUCE is sponsored by the OSD and managed by the AFRL Rome Labs. Please visit <http://www.sprucecommunity.org> to learn more and get your account.



Booz | Allen | Hamilton



Raytheon

SPRUCE

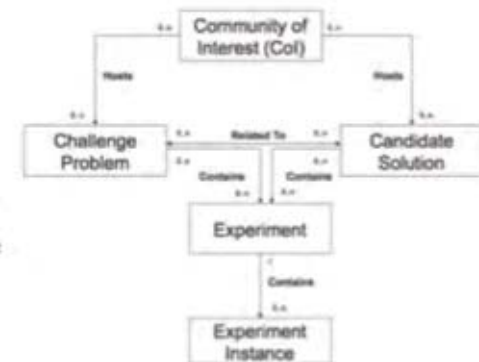
Revolutionizing the way that systems and software engineering technologies are identified, developed and evaluated

Features

- Customized collaboration portal
- Dedicated on-demand experimentation facility
- Repository of realistic problem artifacts
- Experiments to consistently reproduce problems and demonstrate solutions

Repository Elements

- **Community of interest:** virtual team of experts and practitioners to identify problems and work on solutions
- **Challenge problems:** sanitized versions of realistic problems and data that occur on DoD acquisition programs
- **Candidate solutions:** describe proposed solutions to challenge problems
- **Experiments and experiment instances:** showcase challenge problems and evaluate solutions



SPRUCE Can Help ...

Program Engineers

- Discover and reach out to a broad community that can relate to the same technical problems
- Learn about technologies relevant to the problem
- Engage a community of experts to solve the problem
- Make available sanitized, at-scale program data and demonstrate specific problems in the experimentation facility

Software Researchers

- Gain key insights into DoD problems
- Access a repository of real-world problems and representative data
- Quickly and easily engage an active practitioner community
- Effectively demonstrate the technologies and tools in the experimentation facility using program data

Post Challenge Problems



For best results, include at-scale, sanitized data, and reproducible experiments

Experiment



SPRUCE Collaborate



Demonstrate Solutions



For best results, fully use the provided data and reproducible demonstrations

Visit www.sprucecommunity.org to learn more and get an account

A.7 SPRUCE Datasheet
[8 pages]

**Systems and Software Producibility
Collaboration and Experimentation Environment**

>> SPRUCE <<

**Collaborative Systems and Software Engineering
Challenge Problems and Solutions**



***The “go to” place for computer science
challenge problems, artifacts, solutions
and experiments.***

What is SPRUCE?

SPRUCE is an open, virtual, collaborative research and development environment that brings together researchers, developers, practitioners, domain experts and program managers to collaborate around systems and software engineering challenges.

Through well articulated and bounded challenge problems, **SPRUCE** ignites scientific and engineering innovation in systems and software engineering and to help bridge the gap between technology users and technology providers.

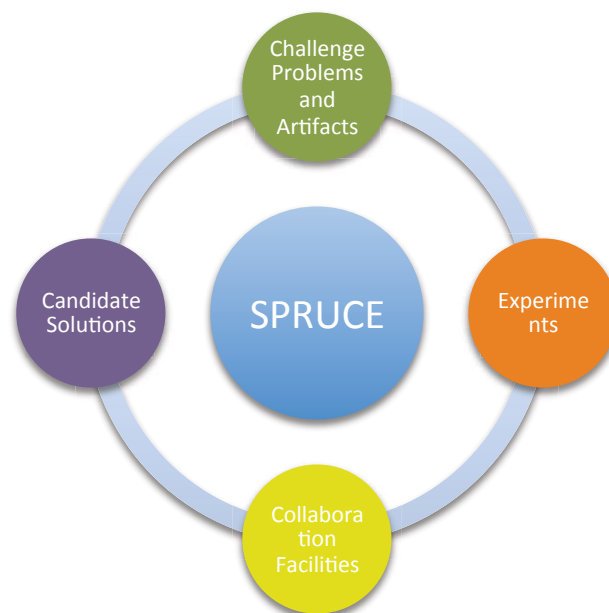
SPRUCE is implemented in a custom web portal to facilitate SPRUCE users access to:

- **Killer Content** of dynamically evolving challenge problems supported by realistic artifacts.
- **Candidate Solutions** to tackle the most demanding challenges.
- An **Experimentation Testbed** built-in, dedicated to test and demonstrate the efficacy of proposed solutions (tools, methods, techniques and run-time technologies) to tackle the specified challenges.
- **Self-organizing Communities of Interest** to leverage the synergy of shared ideas and inspire collaboration.

Rapid advances in systems and software technologies require equally rapid responses in producibility research, techniques, tools and processes



The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.



The SPRUCE project was established under the guidance of the Office of Secretary of Defense (OSD) and executed by the Air Force Research Laboratory Information Directorate (AFRL/RI).

Challenge Problems and Artifacts

Challenge Problems

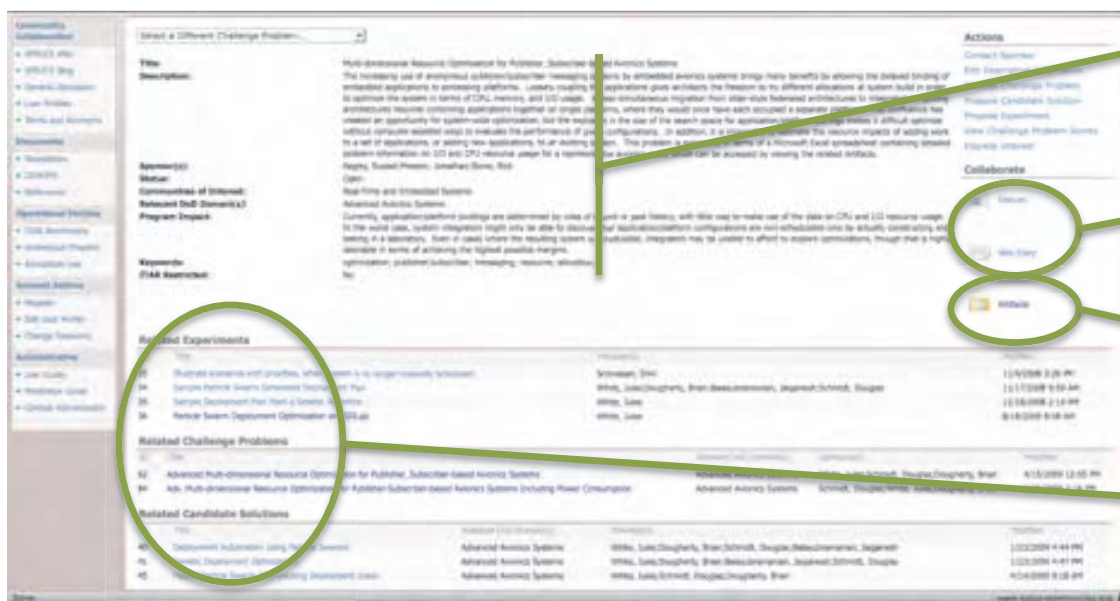
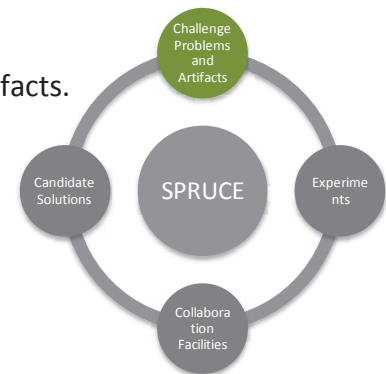
Critical pieces of **SPRUCE**'s killer content are the challenge problems and artifacts.

Challenge problems describe:

- An overall systems or software engineering problem (e.g., Cache false-sharing in multi-core architectures)
- A specific need (e.g., analysis tools to detect the problem)
- Metrics used to evaluate success
- Realistic artifacts

There are more than twenty-five challenge problems in **SPRUCE**, and the number continues to increase.

From a challenge problem's main web page in **SPRUCE**, users can access its artifacts, related challenge problems, proposed solutions and experiments. Users can also browse and participate in community collaborations through dedicated wiki and discussion boards.



Challenge problem description

Wiki and discussion boards

Artifacts

Related challenge problems, solutions and experiments

Artifacts

Artifacts provide real-world context for challenge problems. They represent actual or sanitized data used to demonstrate one or more challenges. These at-scale representations of program data are also used to show convincing evidence that proposed solutions indeed tackle the challenges.

Artifacts in **SPRUCE** represent systems and software engineering data – code, designs, requirements, tests, etc. There are hundreds of artifacts in **SPRUCE**, and new artifacts are being added regularly.

Check out **SPRUCE** for challenge problems and artifacts in technology areas including multi-core, model-driven architecture and design, code analysis, software evolution and system-of-systems

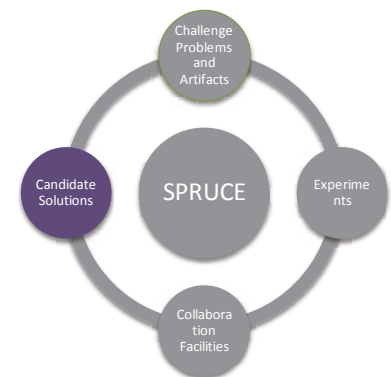
Go to <http://www.sprucecommunity.org> to post a challenge problem

Candidate Solutions

Candidate solutions describe potential solutions to systems and software engineering challenges. They use challenge problem artifacts as context to demonstrate the applicability of the proposed solutions.

A candidate solution's main web page in **SPRUCE** can act as a landing page for the solution technology. Solutions can be described in detail and demonstrated using the **SPRUCE** experimentation facility. Users can also access related solutions, related experiments, and related challenge problems, and can also browse and participate in community collaborations through dedicated wiki and discussion boards.

Today, we have a small (and growing) number of solutions in **SPRUCE**.



Candidate Solution Details

Select a different Candidate Solution...

Title: Deployment Automation Using Particle Swarms
Description: Particle Swarm Optimization (PSO) is a search technique for finding solutions to both linear and non-linear optimization problems. We have a candidate solution that uses a modified PSO algorithm to find deployment solutions that: 1) respect real-time scheduling constraints, 2) respect any additional resource constraints, 3) supports multiple network topologies, and 4) attempts to minimize the end-to-end network bandwidth.
Provider(s): Whites, Jurek, Dougherty, Brian, Schmidt, Douglas, Basudraman, Jagannath
Status: Ready for Experimentation
Solution Type: Software Test/Methodology and Process Recommendations
Commonplace of Interest: Real-Time and Embedded Systems
Relevant DSD Domain(s): Advanced Avionics Systems
Operating System: Platform-independent / Java
Software Platform Details: The solution is implemented in Java. The tool is platform-independent.
Hardware Platform Details: N/A
Licensing Conditions: Apache License v2
Keywords: Multi-resource, deployment, optimization, analysis, real-time scheduling, particle swarm optimization
STAR Restricted: No

Actions:
Candidate Provider
Full Description/Information
Proposed Candidate Solution
Download
Wiki
Discussion
Collaborate
Artifacts

Related Candidate Solutions

Title	Provider(s)	Status
40 Deployment Automation Using Particle Swarms	Advanced Avionics Systems	Ready
41 Semantic Deployment Optimization	Advanced Avionics Systems	Ready
42 Hybrid Particle Swarms / Metaheuristic Deployment Optimization	Advanced Avionics Systems	Ready

Related Experiments

Title	Provider(s)	Status
34 Sample Particle Swarms Generated Deployment Solutions	Whites, Jurek, Dougherty, Brian, Basudraman, Jagannath, Schmidt, Douglas	Completed
35 Particle Swarms Deployment Optimization on Java	Whites, Jurek	Completed

Related Challenge Problems

Title	Provider(s)	Status
36 Multi-dimensional Resource Optimization for Publisher-Subscriber-based Avionics Systems	Advanced Avionics Systems	Ready

Candidate solution description

Wiki and discussion boards

Solution and license or simply a pointer

Related challenge problems, solutions and experiments

Licensing and Intellectual Property

SPRUCE does not impose any licensing approach on the solution provider. They can simply choose to provide a pointer to how an user can access the solution, thus leaving intellectual property in the control of the solution provider.

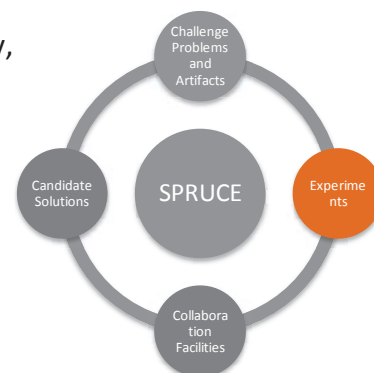
A solution to the multi-dimensional resource optimization challenge problem in SPRUCE led to a funded technology insertion experiment.

Go to <http://www.sprucecommunity.org> to post a solution

Experiments

SPRUCE has a state-of-the-art, easy-to-use, on-demand experimentation facility, based on the proven Emulab technology (<http://www.emulab.net>). This experimentation facility reproduces challenge problems and demonstrates and evaluates the effectiveness of candidate solutions.

SPRUCE users can access this experimentation facility remotely, dynamically request and receive experiment resources, quickly setup experiment configurations, load specific operating systems and software images, conduct experiments, and collect and display experimental results.



SPRUCE provides seamless integration of the experimentation environment and the collaboration environment, allowing reuse, cataloging, and automation of many of these functions.

Once setup, experiments can be parameterized and run repeatedly from a friendly interface, with the results automatically extracted and made available at the **SPRUCE** portal. Results can also be formatted for presentation using **SPRUCE** wikis and artifact libraries. The sample below illustrates experimental results from executing a multi-dimensional resource allocation algorithm in **SPRUCE** . New testbed hardware can easily be added at any time.



In addition to reconfigurable computing blades, the SPRUCE experimentation facility also contains multi-core processor hardware for behavior analysis, performance evaluation, and testing cache and software migration issues associated with multi-core architectures.

Go to <http://www.sprucecommunity.org> to post an experiment

Collaboration

Support for collaboration is a key and essential feature of **SPRUCE**. The ever-growing user population consists of an international mix of participation from academia, industrial research organizations, small businesses and government laboratories and institutions.

SPRUCE users discover each other through affinity searches, join and participate in communities of interest, and use collaboration facilities like wikis and discussion boards.

Affinity Search

With affinity search, users can discover people and publications related to content posted in **SPRUCE**. This allows, for example, challenge problem providers to actively identify and seek out suitable researchers who have the skills to develop solutions to these challenges. Using search keywords that are automatically extracted from unstructured text, public databases such as Citeseer are searched for matches and rank-ordered results listing publications and authors are displayed.

Communities of Interest

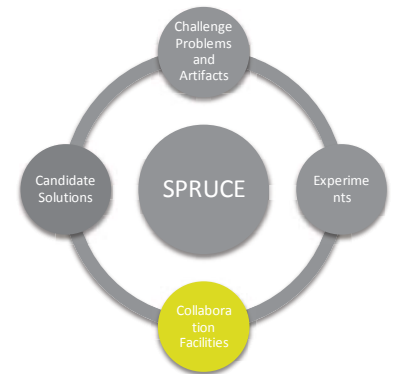
Communities of Interest (COI) naturally organize **SPRUCE** content (i.e., challenge problems, candidate solutions and associated discussions) around broad, focused topic areas. They also serve as virtual meeting places for **SPRUCE** users. Current sample COI are: real-time/embedded systems, multi-core architectures and feature oriented software analytics.

Wikis and Discussion Boards

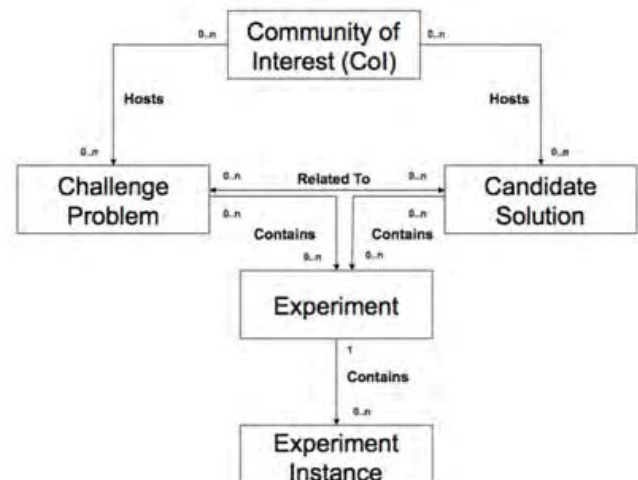
SPRUCE provides individual workspaces in the form of wikis, discussion boards and artifact libraries for each Challenge Problem, Candidate Solution and Experiment. In the very near future, look for integration with Facebook, Twitter and LinkedIn.

Structured Collaboration

Collaboration is structured around the key concepts of **SPRUCE** (challenge problems, candidate solutions and experiments) to facilitate engagement. These concepts are simple to understand yet simultaneously broad enough to apply to any domain while still allowing in-depth exploration of a wide variety of topic areas.



Check out and participate in collaborations around the Model Driven Architecture & Design (MDA&D) grand challenge or any of its constituent challenge problems.



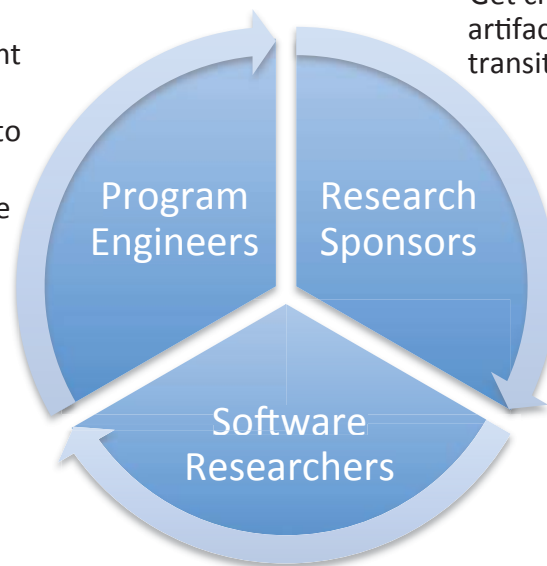
Visit <http://www.sprucecommunity.org> to collaborate with others

SPRUCE Benefits

SPRUCE provides valuable benefits to a wide variety of communities, including program engineers, software researchers and research sponsors.

As a **Program Engineer**, you can

- Discover and reach out to a broad community that can relate to the same technical problems.
- Learn about technologies relevant to the problem.
- Engage a community of experts to solve the problem.
- Make available sanitized, at-scale program data and demonstrate specific problems in the experimentation facility.



As a **Research Sponsor**, you can

- Access a continuously evolving virtual community of people, problems and technologies.
- Get critical insights, data and artifacts to justified, start, and/or transition your program!

As a **Software Researcher**, you can

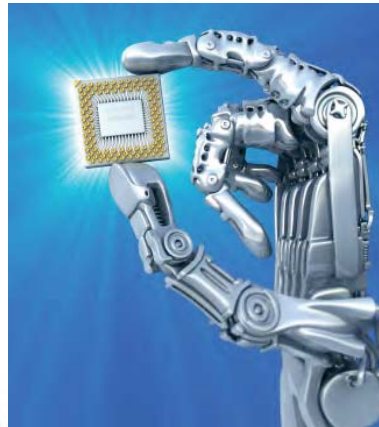
- Gain key insights into domain problems.
- Access a repository of real-world problems and representative data.
- Quickly and easily engage an active practitioner community.
- Effectively demonstrate the technologies and tools in the experimentation facility using program data.

Originally launched to support research, collaboration and experimentation for large-scale U.S. Department of Defense (DoD) systems, **SPRUCE** has expanded its scope to include all domains of computer science. International participation is both solicited and encouraged.

A Call for Participation

Our goal is to make **SPRUCE** the “go to” virtual place for computer science challenge problems, artifacts, solutions and experiments. We need your help to make this goal a reality.

Become involved today. Visit <http://www.sprucecommunity.org>



SPRUCE

*The “go to” place for computer science
challenge problems, artifacts, solutions
and experiments.*

For more information, please visit: <http://www.sprucecommunity.org>

A.8 *SPRUCE Brochure*
[4 pages]

SPRUCE

www.SpruceCommunity.org

Real Problems. Real Data. Real Solutions.
A Virtual Laboratory for the Computer Science Community

JOIN TODAY



SPRUCE was established as a resource for the computer science community
by the Office of the Secretary of Defense.

Imagine



...a web portal dedicated to capturing real challenges, real data and real solutions in computer science and systems and software engineering.

...a **virtual experimentation facility** that acts as a proving ground to validate challenges and test solutions.

...a **collaborative community** intent on bringing forth and solving the toughest challenges.

That's SPRUCE!



...an **online virtual laboratory** for advancing computer science research and development. It is free and openly available to all participants.

SPRUCE is a resource for the computer science community. Program engineers, practicing technicians, university researchers and research sponsors can exchange ideas and collaborate to overcome current and emerging computer science challenges of the 21st century.

Challenge Problems and Artifacts

The Killer Content

SPRUCE empowers you, the software developer or systems integration engineer, to post computer science challenges that entice the best and brightest researchers to work on them.

These challenges may represent:

- Known or perceived systems or software engineering problems (e.g., cache false-sharing in multi-core architectures).
- A specific need (e.g., an analysis tool to detect buffer overflow).
- Computer science grand challenges (e.g., integrated model-based engineering across the product lifecycle).

SPRUCE also contains realistic artifacts — code, designs, requirements, tests, etc. — that represent actual or sanitized data that provide context for challenges.



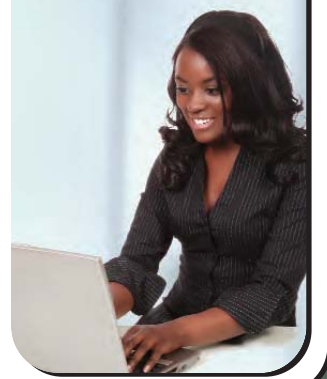
Join Today at SpruceCommunity.org

Candidate Solutions | The Promise of Solutions Motivates Challenges

For the passionate computer science researcher, SPRUCE provides:

- Access to a rich portfolio of challenges and readily usable artifacts to provide a context for developing truly novel solutions.
- A place to post your solutions, having far-reaching implications for the industry.

Success Story: A solution from a university to the multi-dimensional resource optimization challenge problem in SPRUCE led to a partnership between a large company (problem provider) and the university (solution provider), followed by a technology insertion experiment.



Experimentation Testbed | Experiment | Reproduce | Test

SPRUCE provides:

- Remote access to a state-of-the-art, easy-to-use, on-demand experimentation facility to demonstrate challenge problems and evaluate the effectiveness of candidate solutions.
- Rich interaction with the facility, enabling quick, dynamic experiment setup and execution, and an intuitive means to collect experimental data and share experimental results.



Access to the SPRUCE experimentation testbed is currently free to registered users.

Collaboration | Meet and Team with the Experts

Collaboration is a key and essential feature of SPRUCE.

SPRUCE structures collaboration around communities of interest that:

- Organize naturally around focused topic areas.
- Serve as a convenient virtual meeting place for **SPRUCE** users.

Can't find the computer science topic you need to discuss?

Setup and recruit participation in your own community of interest!

SPRUCE's ever-growing population consists of participation from universities, industry, small businesses and government laboratories and institutions.



Join Today at SpruceCommunity.org

I am part of SpruceCommunity.org



Systems and software practitioners

- Discover and reach out to a broad community that can relate to the same technical problems.
- Engage a community of experts to solve problems.
- Learn about technologies relevant to your own challenges.



Researchers / solution providers

- Access a repository of real-world problems and representative data.
- Quickly and easily engage an active practitioner community.
- Demonstrate technologies and tools in the experimentation testbed.
- Engage a community of experts to solve problem.
- Learn about technologies relevant to your own challenges.



Research sponsors

- Access a continuously evolving virtual community of people, problems and technologies.
- Get critical insights, data and artifacts to justify, start, and/or transition your program.



College / university participants

- Find interesting, challenging and relevant computer science research topics and teaching materials.
- Build relationships with academic, industry and government partners.

SPRUCE gives users the flexibility to protect their intellectual property, as needed.

A Call for Participation

Come join our vibrant **SPRUCE** community. Help make **SPRUCE** the “go to” virtual laboratory for computer science challenge problems, artifacts, solutions and experiments.



Join Today at SpruceCommunity.org

A.9 SPRUCE Information Pamphlet
[8 pages]

SPRUCE

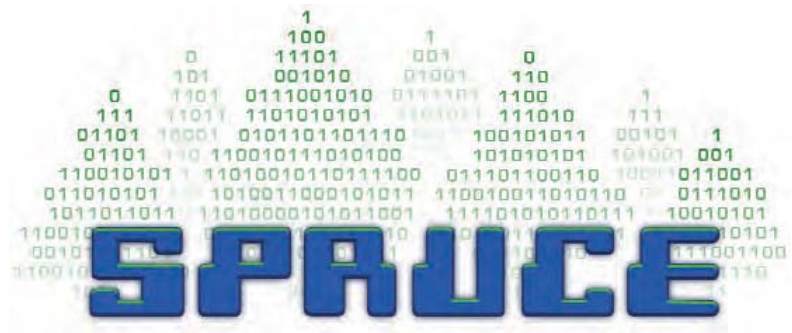
www.SpruceCommunity.org

Real Problems. Real Data. Real Solutions.
A Virtual Laboratory for the Computer Science Community

JOIN TODAY



SPRUCE was established as a resource for the computer science community
by the Office of the Secretary of Defense.



www.SpruceCommunity.org

Information Pamphlet

Table of Contents

■ What is SPRUCE?	pg. 3
■ Challenge Problems and Artifacts	pg. 4
■ Candidate Solutions	pg. 5
■ Experiments	pg. 6
■ Collaboration	pg. 7
■ SPRUCE Benefits	pg. 8



What is SPRUCE?

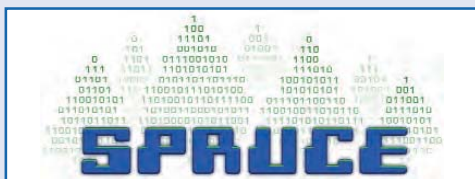
The Systems and Software Producibility Collaboration & Experimentation Environment

SPRUCE is an open, online virtual laboratory for advancing computer science R&D. SPRUCE brings together researchers, developers, practitioners, domain experts and program managers. SPRUCE allows these users to collaborate on computer science, systems and software engineering and producibility challenges.

SPRUCE ignites innovation through well articulated and bounded challenge problems in computer science applications, supported by a rich collection of data and artifacts, to move computer science from theory to producible, useable applications.



SPRUCE is the place to go to advance computer science.

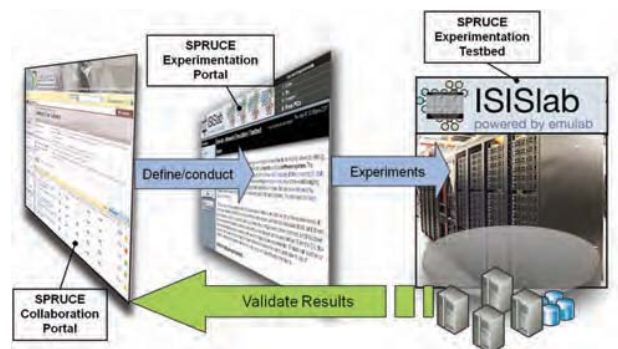


Rapid advances in systems and software technologies require equally rapid responses in computer science research, especially as it relates to producibility research, techniques, tools and processes.

The SPRUCE Portal and Experimentation Testbed

SPRUCE is implemented as a custom web portal that enables SPRUCE user to:

- Post dynamically evolving **Challenge Problems** supported by realistic **Artifacts**.
- Propose **Candidate Solutions** to resolve those perplexing and demanding challenges.
- Validate challenges and test solutions using the built-in **Experimentation Testbed**.
- Create, join and participate in self-organizing **Communities of Interest** that inspire collaboration and leverage the synergy and power of the shared intellect of the community.



Get involved. Visit SpruceCommunity.org

The SPRUCE project was established under the guidance of the Office of Secretary of Defense (OSD) and executed by the Air Force Research Laboratory Information Directorate (AFRL/RI).



THE “KILLER CONTENT”

Challenge Problems & Artifacts

Pose Your Most Vexing Challenge Problems

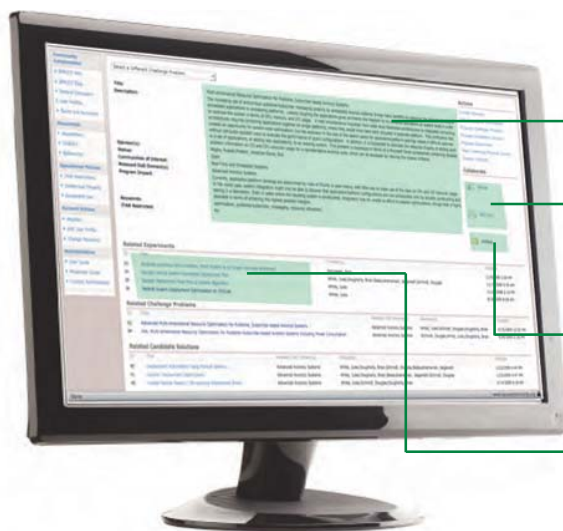
SPRUCE empowers you, the software developer or systems integration engineer, to post computer science challenges that entice the best and brightest researchers to work on them.

Challenge the broad, diverse, intellectual resources of the SPRUCE community with:

- Known or perceived systems or software engineering problems (e.g., cache false-sharing in multi-core architectures).
- A specific need (e.g., an analysis tool to detect buffer overflow).
- Computer science grand challenges (e.g., integrated model-based engineering across the product lifecycle).

Visit SPRUCE to view the full suite of challenge problems.

Check out **SPRUCE** for challenge problems and artifacts in technology areas including multi-core, multi-modeling and model integration, and cyber physical systems.



Challenge problem description

Wiki and discussion boards

Artifacts

Related challenge problems, solutions and experiments

Go to www.SpruceCommunity.org to see examples in detail.

Artifacts Provide the Context

Artifacts provide **real-world context** for challenge problems. They represent actual or sanitized data to showcase one or more challenges and provide convincing evidence that proposed solutions indeed resolve challenges. Artifacts in SPRUCE take on a variety of forms – code, designs, requirements, tests, etc.

There are hundreds of artifacts in **SPRUCE**, and the number is growing.

Go to www.SpruceCommunity.org to post a challenge



THE PROMISE OF SOLUTIONS MOTIVATES CHALLENGES

Candidate Solutions

Solve Those Vexing Challenge Problems:

Candidate solutions describe potential solutions to computer science challenges — challenge problems. They draw context from associated artifacts and can be demonstrated using the SPRUCE experimentation facility.

View other candidate solutions, related challenge problems and artifacts, and related experiments.

Review the ever increasing number of community-offered solutions.

In SPRUCE, candidate solutions tap into the rich, collective resources of the community intellect, demonstrating their ability to resolve problems, and lowering the barrier to successful transition into practice.

A solution to the multi-dimensional resource optimization challenge problem in SPRUCE led to a partnership between a large company (problem provider) and the university (solution provider), followed by a technology insertion experiment.



Candidate solution description

Wiki and discussion boards

Solution and license or simply a pointer

Related challenge problems, solutions and experiments

Go to www.SpruceCommunity.org to see examples in detail.

Licensing and Intellectual Property Protection

SPRUCE does not impose any particular licensing approach or intellectual property protection mechanism on the challenge problem posters or solution providers. Users can select from a list of options, or define their own mechanisms.

Go to www.SpruceCommunity.org to post a solution



EXPERIMENT | REPRODUCE | TEST

Experiments

Use the SPRUCE Experimentation Testbed to Verify Challenges and Validate Solutions

Use the **SPRUCE** experimentation facility from the convenience of your workstation to dynamically request and receive experiment resources, quickly setup experiment configurations, load specific operating systems and software images, conduct experiments, and collect and display experimental results.

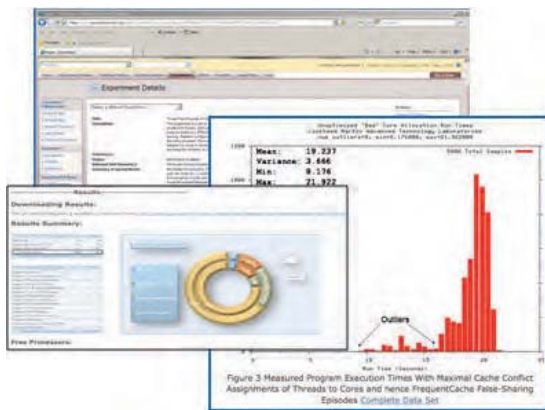
Parameterize and run test repeatedly from a user-friendly interface.

Work with the SPRUCE team to **add new or specialized hardware**.

View results, automatically extracted and posted on the SPRUCE portal.

Format results for presentation using **SPRUCE wikis and artifact libraries**.

*The illustration below shows formatted results from a multi-dimensional resource allocation algorithm in **SPRUCE**.*



Go to www.SpruceCommunity.org to see examples in detail.

SPRUCE provides seamless integration of the experimentation testbed and the collaboration environment, simplifying interaction among technical and non-technical users.



SPRUCE has an easy-to-use, state-of-the-art, on-demand experimentation facility to reproduce challenge problems and evaluate the effectiveness of candidate solutions.



In addition to reconfigurable computing blades, the **SPRUCE** experimentation facility also contains multi-core processor hardware for behavior analysis, performance evaluation, and testing cache and software migration issues associated with multi-core architectures.

Go to www.SpruceCommunity.org to post an experiment



MEET AND TEAM WITH EXPERTS

Collaboration

The Power of the Collective Intellect Multiplied

Support for collaboration is a key and essential feature of **SPRUCE**. The ever-growing user population consists of an international mix of participants from academia, industrial research organizations, small businesses and government laboratories and institutions.

SPRUCE users discover each other through affinity searches, join and participate in communities of interest, and use collaboration facilities like wikis and discussion boards.

Affinity Search

With **Affinity Search**, users can discover people and publications related to content posted in **SPRUCE**. This allows, for example, challenge problem providers to actively identify and seek out suitable researchers who have the skills to develop solutions to these challenges. Searching for keywords that are automatically extracted from unstructured text, public databases — such as Citeseer — are searched for matches and rank-ordered results listing publications and authors are displayed.

Communities of Interest

Communities of Interest (COI) naturally organize around **SPRUCE** content (*i.e.*, *challenge problems*, *candidate solutions* and *associated discussions*) and focused topic areas. They also serve as virtual meeting places for **SPRUCE** users. Current sample COI are: real-time/embedded systems, multi-core architectures and feature oriented software analytics. Can't find the topic you need to discuss — set up, recruit and build your own COI.

Wikis and Discussion Boards

SPRUCE provides individual workspaces in the form of wikis, discussion boards and artifact libraries for each Challenge Problem, Candidate Solution and Experiment. **SPRUCE** is also accessible through Facebook, Twitter, and LinkedIn. Go to **www. SpruceCommunity.org** to learn more.

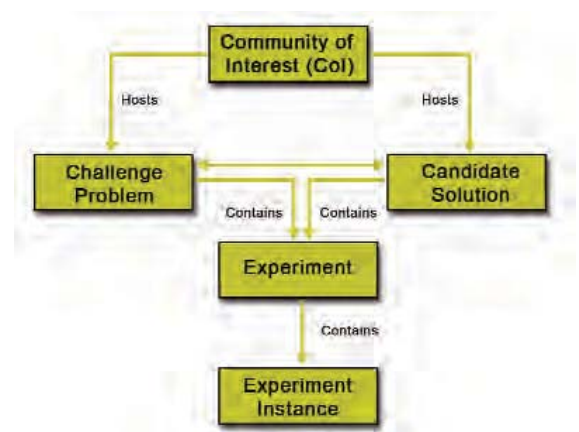
Structured Collaboration

Collaboration is structured around the key concepts of **SPRUCE** (*challenge problems*, *candidate solutions* and *experiments*) to facilitate engagement.

These concepts are simple to understand yet simultaneously broad enough to apply to any domain while still allowing in-depth exploration of a wide variety of topic areas.



Check out and participate in collaborations in **SPRUCE**, such as the Model Driven Architecture & Design (MDA&D) grand challenge or any of its constituent challenge problems.



Visit **www.SpruceCommunity.org** to collaborate with others

SPRUCE Benefits

SPRUCE provides valuable benefits to a wide variety of communities, including program engineers, software researchers, college/university members and research sponsors.

Systems and software practitioners



Discover and reach out to a broad community that can relate to the same technical problems.

Engage a community of experts to solve problems.

Learn about technologies relevant to your own challenges.

Researchers / solution providers



Access a repository of real-world problems and representative data.

Demonstrate technologies and tools in the experimentation testbed.

Engage a community of experts to solve problems, or learn about technologies relevant to your own challenges.

College / university participants



Find interesting, challenging and relevant computer science research topics and teaching materials.

Build relationships with academic, industry and government partners.

Research sponsors



Access a continuously evolving virtual community of people, problems and technologies.

Get critical insights, data and artifacts to justify, start, and/or transition your program.

Originally launched to support research, collaboration and experimentation for large-scale U.S. Department of Defense (DoD) software and systems engineering, SPRUCE has expanded its scope to include all domains of computer science. International participation is both solicited and encouraged.

A Call for Participation

Come join our vibrant **SPRUCE** community. Help make **SPRUCE** the “go to” virtual laboratory for computer science challenge problems, artifacts, solutions and experiments.

Become involved today. Visit www.SpruceCommunity.org

List of Acronyms

Advance Software Engineering Technologies for the Software Producibility Initiative (ASETS)
Air Force Research Laboratory (AFRL)
Birds-of-a-Feather (BoF)
Broad Agency Announcements (BAAs)
Collaboration Technologies symposium (CTS)
Communities of Interest (CoI)
Concept of Operations (CONOPS)
Cyber Innovation Center (CIC)
Cyber Security & Information Systems Information Analysis Center (CSIAC)
Cyber-Physical Systems Virtual-Organization (CPS-VO)
Data and Analysis Center for Software (DACS)
Department of Defense (DoD)
Distributed Real-time and Embedded (DRE)
General Accounting Office (GAO)
Intellectual Property (IP)
International Council on Systems Engineering (INCOSE)
International Traffic in Arms Regulations (ITAR)
Lockheed Martin Aeronautics (LM Aero)
mobile ad hoc network (MANET)
Model Based Systems Engineering (MBSE)
National Institute of Standards and Technology (NIST)
National Research Council (NRC)
National Science Foundation (NSF)
Office of the Secretary of Defense (OSD)
Out-of-the-Box (OOTB)
Particle Swarm Optimization (PSO)
Real-Time Systems Symposium (RTSS)
Return on Investment (ROI)
Service-Oriented Architectures (SOA)
Software and Systems Test Track (SSTT)
Software Assurance Metrics And Tool Evaluation (SAMATE)
Software Engineering Institute (SEI)
Systems and Software Producibility Collaboration and Experimentation Environment (SPRUCE)
Systems and Software Technology Conference (SSTC)
Vanderbilt University (VU)
Warfighter-Industry Collaboration Enterprise (WIC-E)